

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMÁTICA



Plataforma de gestión de retos gamificados y analíticas

CARLOS MARRÓN DÍAZ
JORGE MERINO MARTÍNEZ

TRABAJO DE FIN DE GRADO EN INGENIERÍA DEL SOFTWARE
DIRIGIDO POR: IVÁN MARTÍNEZ ORTIZ

SEPTIEMBRE DE 2018

En primer lugar, quiero dar las gracias a mi padres, José y Rosa, por haberme dado su apoyo todos estos años y haberme dado fuerzas día a día hasta que por fin he terminado. A mi tía Magdalena, por haber confiado en mí sin dudar en ningún momento y darme siempre ánimos. A mi novia, Ana, por haber estado conmigo y haberme aguantado en todos los momentos difíciles que he tenido este año, y haberme hecho creer que todo era posible. Y a nuestro director, Iván Martínez, por su ayuda, su labor docente y su compromiso con el proyecto.

Carlos Marrón Díaz.

A mi familia y amigos, por todo el apoyo y el ánimo que me han ofrecido a lo largo del transcurso de la carrera. A todos los compañeros y amistades que he hecho durante esta etapa y al profesorado de la Facultad de Informática. También agradecer a Iván Martínez Ortiz, director de proyecto, por la ayuda aportada. Y a la Universidad Complutense, por ofrecerme esta experiencia académica, en la que he podido crecer, aprender y formarme, durante todos estos años.

Jorge Merino Martínez.

Resumen

El objetivo principal de este Trabajo de Fin de Grado, ha sido el desarrollo de una herramienta integrada y abierta para la gestión de retos, incluir elementos de gamificación y recolección de analíticas. La solución resultante propone una arquitectura modular y, para cada uno de los principales componentes, proporciona o bien una implementación de referencia para el módulo o bien reutilizando una aplicación existente para implementar la funcionalidad deseada. El resultado es un prototipo usable, completo y funcional de la arquitectura propuesta.

Palabras clave: Retos, Gamificación, Analíticas.

Abstract

The main objective of this work, has been the development of an open and integrated platform for that allows users to manage challenges, to include gamification aspects and to collect analytics. The final solution proposes a modular architecture and, for each of the modules, includes an ad-hoc reference implementation or integrates an existing solution. The actual result is an usable, complete and functional prototype of the proposed architecture.

Keywords: Challenges, Gamification, Analytics.

Autorización de difusión

Los abajo firmantes, alumno/s y tutor/es del Trabajo Fin de Grado (TFG) en el Grado en Ingeniería del Software de la Facultad de Informática, autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores este Trabajo de Fin de Grado cuyos datos se detallan a continuación. Así mismo autorizamos a la Universidad Complutense de Madrid a que sea depositado en acceso abierto en el repositorio institucional con el objeto de incrementar la difusión, uso e impacto del TFG en Internet y garantizar su preservación y acceso a largo plazo.

Nombre de los alumnos:

- Carlos Marrón Díaz
- Jorge Merino Martínez

Nombre del tutor:

- Iván Martínez Ortiz

Fdo. Carlos Marrón Díaz

Fdo. Jorge Merino Martínez

Fdo. Iván Martínez Ortiz

Índice

Resumen.....	VII
Abstract	VIII
Autorización de difusión.....	IX
Índice	1
1. Introducción.....	5
1.1. ¿Qué es un reto?	6
1.1.1. Tipos de reto	6
1.1.2. Herramientas educativas basadas en retos	7
1.1.3. Resumen comparativo de Quizziz y Kahoot!	13
1.2. ¿Qué es la gamificación?	14
1.2.1. Motores de gamificación.....	17
1.3. Analíticas	18
1.3.1. Herramientas de analíticas de aprendizaje.....	22
2. Objetivos de este trabajo	22
3. Plan de trabajo	24
3.1. Metodología de desarrollo.....	24
3.2. Planificación del proyecto	25
4. Resultados.....	27
4.1. Evolución del desarrollo del proyecto y descripción de la solución final integrada.....	27
4.1.1. Motor de retos.....	27
4.1.2. Motor de gamificación	28
4.1.3. Analíticas	29
4.1.4. Solución final integrada.....	29
4.2. Arquitectura del software de Plataforma RGA.....	31
4.2.1. Modelo del dominio	32
4.2.2. Diagrama de clases de negocio.....	33

4.2.3. Diagrama de clases de presentación.....	34
4.3. Arquitectura del software de Quiz	36
4.3.1. Modelo del dominio	36
4.3.2. Diagrama de clases de negocio.....	37
4.3.3. Diagrama de clases de presentación.....	38
4.4. Tecnologías utilizadas	39
4.5. Fases del desarrollo.....	41
4.5.1. Fase de planificación	41
4.5.2. Fase de prototipado	42
4.5.3. Fase de implementación	43
4.5.4. Fase de integración	45
4.5.5. Implementación e integración de Quiz	46
4.5.6. Implementación del módulo de analíticas.....	47
4.6. Resultados finales	50
5. Aportaciones personales	51
5.1. Carlos Marrón Díaz.....	51
5.2. Jorge Merino Martínez	53
6. Conclusiones	55
7. Conclusions	56
8. Trabajo futuro	57
9. Future work	58
Bibliografía.....	59
Referencias.....	63
Anexo I: Manual de instalación	65
Anexo II: Manual de usuario	69
Manual del profesor.....	69
Manual del alumno.....	76
Manual del profesor, Quiz	80

Manual del alumno, Quiz	85
Anexo III: Referencia de la API de Plataforma RGA.....	91
Anexo IV: Referencia de la API de Quiz	95

1. Introducción

La motivación de los estudiantes representa un aspecto clave en el proceso de aprendizaje [1] [2]. En este sentido, los autores Malone y Lepper proponen una teoría sobre la motivación basada en seis características que permiten identificar si una actividad es motivante [3]: el reto, la curiosidad, el control, la cooperación, la competición y el reconocimiento.

Por otro lado, el uso de la tecnología ha tenido un gran impacto en el proceso de aprendizaje precisamente al ser usadas como mecanismos o herramientas motivantes, entre otras, el uso de los juegos serios [4], la aplicación de estrategias de gamificación [5] y los sistemas de respuesta en clase [6].

Dentro de la categoría de los sistemas de respuesta en clase, las herramientas Kahoot! [7] y Quizizz [8] destacan sobre el resto debido a su facilidad de uso y a que son ampliamente utilizadas en los diferentes niveles educativos. Estas herramientas, pese a ser consideradas como un sistema de respuesta en clase, también incluyen características de gamificación y competitividad.

Las herramientas como Kahoot!, Quizizz o los juegos serios no sólo son utilizados como mecanismo de motivación en clase, sino que pueden aprovecharse también como mecanismos de evaluación de los estudiantes. En este sentido, todas estas herramientas al haber sido concebidas en el ámbito de la educación incluyen algún mecanismo de evaluación, por ejemplo, la puntuación obtenida en el test o reto (Kahoot!, Quizizz) o el número de retos superados o puntuación obtenida en el juego serio. Sin embargo, además de estas métricas simples también es posible recopilar otros eventos de interés que puedan ser de interés en el proceso de aprendizaje, por ejemplo, si el reto ha sido completado con éxito o no, el tiempo necesario en completar el reto, etc. Esta información puede ser utilizada con varios objetivos, por ejemplo, la validación de efectividad o el diseño de la propia herramienta o como mecanismo de evaluación o apoyo a la evaluación de los alumnos.

En ese sentido, han surgido un nuevo campo de investigación alrededor de la recopilación y análisis de los eventos más relevantes (desde el punto de vista educativo) que habitualmente generan las herramientas educativas. Este nuevo campo de investigación recibe el nombre de analíticas de aprendizaje [9] y puede definirse como la “medición, recolección, análisis y generación de informes en base a los datos de los estudiantes y su contexto educativo, con el propósito de comprender y optimizar el aprendizaje y los contextos donde éstos ocurren” [10].

El uso de herramientas de participación / respuesta en clase y la aplicación de técnicas de gamificación como mecanismos de mejora de la motivación y participación en clase, además del uso de las analíticas de aprendizaje para comprender y mejorar la aplicación de los retos o herramientas de participación en clase, son los elementos sobre los que se construyen el trabajo realizado en este trabajo de fin de grado. El resto del capítulo realiza una introducción, no exhaustiva, de estos elementos fundamentales en los que se basan las herramientas desarrolladas como parte de este trabajo.

1.1. ¿Qué es un reto?

Un reto es un conjunto de preguntas, torneos y/o cuestionarios, con sus respectivas respuestas, centrado en un determinado tema, de cualquier campo o área educativa. Un reto podrá estar dirigido para cualquier rango de edad, ya sea para los cursos más básicos de la enseñanza educativa, como en ámbitos universitarios e incluso laborales. Los retos serán elaborados por la figura de un profesor, coordinador, jefe, etc. y será contestado por los participantes o alumnos objetivos del reto.

Los alumnos intentarán resolver este reto de forma individual o en grupos. Por cada acierto el participante recibirá una puntuación.

Como resultado del reto, el profesor puede obtener datos y estadísticas sobre la evolución de su clase, además de poder conocer si sus alumnos tienen consolidados los conceptos vistos en clase, o cómo se desenvuelven en un determinado tema.

1.1.1. Tipos de reto

Los retos permiten realizar preguntas de diversas formas, haciendo de este modo más fácil captar la atención del alumno.

Entre las distintas formas, podemos diferenciarlos según la forma de respuesta:

- De **respuesta única**, con una única opción que marcar.
- De **múltiples respuestas**, es decir, puede haber más de una respuesta correcta por lo que habría que seleccionar una o varias según sea lo correcto.
- De **respuestas simples**, elegir la opción correcta entre las que aparezcan.
- De **afirmación y negación**, puede ser Si/No o Verdadero/Falso.

- De **respuestas de estimación**, en las que la pregunta pide como respuesta un número y se debe contestar con la cifra correcta. En muchos casos se daría como válido una aproximación predeterminada por el profesor, aunque no se sumaría el 100% de la puntuación puesto que no sería del todo correcto.
- De **secuencias**, sería ir ordenando las respuestas en un orden.
- De **respuesta corta**, en este caso hay que dar una pequeña explicación o escribir la palabra que se crea correcta.
- **Sondeos**, aunque no son muy usados pero también aparecen en algunas aplicaciones.

1.1.2. Herramientas educativas basadas en retos

Las acciones educativas persiguen el aprendizaje y la consecución de determinados objetivos según las necesidades de cada usuario. Pero no todas las acciones consiguen la misma eficacia, o bien por plantear objetivos distintos a las necesidades o bien por una creación de metodología inapropiada.

Para que el proceso educativo resulte atractivo, es necesario crear una metodología o procedimientos llamativos, atractivos y eficaces que promuevan el interés y la motivación de aquellas personas que usen las aplicaciones.

De este modo, se han ido desarrollando aplicaciones de carácter educativo para promover procesos de aprendizaje más interesantes y fuera de la pauta educativa convencional.

Quizizz

Una de las aplicaciones es Quizizz, por medio de la cual se crea el objetivo principal de aprender a través de retos.

Esta aplicación se concibe como inclusiva dentro del aula, puesto que se hace partícipe tanto al alumnado como al profesorado, promoviendo las nuevas tecnologías y el desarrollo de las mismas en el ámbito educativo. Además, tiene la posibilidad de abordar diferentes materias haciendo posible el desarrollo de diferentes conocimientos.

De este modo Quizizz se presenta como innovadora, colaborativa y actual, creando un modo diferente de conocimiento y aprendizaje, accesible a cualquier usuario.



Sitio web: <https://quizizz.com>

La interfaz inicial muestra:

- Los retos más populares de los últimos días.
- Un buscador.
- Una opción para ser redirigido a una página en la cual puede introducir el código dado por el profesor.
- Distintas categorías, donde elegir la que más interesante le resulte para jugar un reto sobre ello.

En esta aplicación, se puede acceder a los retos y realizarlos siempre y cuando en ese mismo momento haya otra persona que realice ese mismo reto. En caso contrario, el usuario entrará en una “sala de espera”. En esta pantalla podrá:

- Esperar hasta que entre algún otro usuario.
- Salir de ella y buscar otros retos en los que haya personas participando.

Quizziz tiene de particular que no es necesario que el profesor muestre una pantalla con las preguntas y las diferentes opciones de respuesta, puesto que cada usuario puede ver en su propio dispositivo las preguntas con sus respuestas correspondientes.

La página principal da la opción tanto de elegir entre distintos idiomas como de cambiar el tema de fondo de pantalla.

Una vez hecho el reto, se crean automáticamente unas estadísticas a partir de los distintos datos que recoge, estos son:

- Número de aciertos y fallos de cada jugador.
- Tiempo medio que se ha usado en realizar el reto.
- Una lista de todas las preguntas del reto indicando en cuales se ha fallado, para poder saber qué conceptos necesita repasar más o hacer más hincapié.

Las opciones iniciales para crear un reto es el nombre, una imagen y el idioma en el cual se va a realizar. Un reto puede ser tanto público como privado. Y las preguntas pueden crearse desde cero, o bien, ser importadas de otros retos ya realizados previamente.

En cuanto a los tipos de pregunta, solo se permiten dos modalidades según sea su respuesta, de opción única o múltiple. Estas respuestas pueden ser de palabras, frases e imágenes y el tiempo que tienen para contestarlas es de entre 5 segundos y 15 minutos.

Además, ofrece la opción de elegir el rango por edad del alumnado para el cual se ha creado el reto. Adaptándolo mejor a las características y al nivel académico de dicho curso escolar dentro del sistema educativo. Asimismo, permite organizar el reto mediante etiquetas o temas y a su vez a un “*topic*” dentro del tema seleccionado.

Otra característica que cabe destacar es la opción de imprimir el cuestionario creado, esto permite que el profesor lo pueda mandar como deberes a realizar en casa, si no hubiese tiempo suficiente para terminar de realizarlo en clase.

Como última característica de esta aplicación se encuentra la posibilidad de añadir memes, que saldrán al fallar o acertar las preguntas en los cuestionarios según el profesorado decida, con el propósito de reforzar la atención del estudiantado y amenizar el aprendizaje.

Kahoot!

Otra de las aplicaciones es Kahoot! cuya meta es el aprendizaje a través de retos. La aplicación fue concebida para el mismo uso que Quizziz, con bastantes características diferenciadas entre ellas, que serán tratadas en el punto [1.1.3. Resumen comparativo de Quizziz y Kahoot!](#), pero con la misma idea principal de facilitar a los profesores la forma de evaluar a los alumnos de forma divertida y entretenida, o incluso utilizándolo de forma de juego entre distintos grupos de participantes.



Para poder entrar en la web se necesita, bien un código de reto, o estar registrado. Para esta última forma de acceso será necesario seleccionar si se está accediendo como profesor, estudiante, por hobby o por trabajo. En caso de elegir la opción laboral, la aplicación ofrece unos días de prueba gratuitos, sin embargo expirado el plazo se deberá adquirir la versión de pago, ya que cuenta con unos ajustes avanzados que el resto de cuentas no poseen.

Sitio web: <https://kahoot.com>

La interfaz inicial muestra:

- Una serie de retos.
- Un buscador.
- Distintas categorías de retos.

Si el usuario fuese un profesor puede crear nuevos retos o usar retos ya existentes en la aplicación. Para que el profesor pueda usar Kahoot! en clase con sus alumnos, se necesita:

1. Un dispositivo desde donde manejar la aplicación, ya sea en su formato web o la versión móvil descargable.
2. Una gran pantalla desde donde todos los alumnos puedan ver cada pregunta con sus respectivas respuestas. Una vez se supere el tiempo de respuesta aparecerá en la pantalla cuál es la respuesta correcta y un diagrama con el número de alumnos que ha seleccionado cada respuesta.
3. Un dispositivo por cada alumno que participe, desde el cual puedan acceder a la aplicación introduciendo el código dado por el profesor. A continuación, escribirán su apodo con el que aparecerán en pantalla. En la pantalla de sus dispositivos solo verán los colores de las diferentes opciones de respuesta.

El profesor al lanzar el reto tiene la opción de elegir si se quiere jugar individualmente o por equipos; así como de activar o desactivar algunas opciones del juego, por ejemplo: dar premios en caso de aciertos consecutivos, cambiar el orden de preguntas aleatoriamente, etc.

Al finalizar, el profesor tiene la opción de descargarse los resultados o guardarlos en Google Drive, el alumno en su caso debe de votar que le ha parecido el reto.

La aplicación se encuentra únicamente en inglés, aunque los diferentes retos pueden crearse en cualquier idioma.

Para crear un reto, la primera opción que nos aparece es el tipo de reto que se desea crear:

- Respuesta simple o múltiple.
- Secuencia.
- Encuestas.

A continuación, se requerirá el nombre, una descripción con palabras clave, una imagen, el idioma en el cual se va a realizar, y el grupo social de gente para la cual se creará el reto. Si se desea también se puede añadir un vídeo de YouTube como introducción.

El reto puede ser tanto público como privado. Las respuestas únicamente pueden ser palabras o frases, el tiempo que tienen para contestarlas es de entre 5 segundos y 2 minutos, dependiendo del tipo de reto que se desee crear.

arsnova.click

arsnova.click, en adelante Arsnova, es una aplicación de código abierto, por lo que cualquier persona puede ver el código, modificarlo y mejorarlo. La aplicación fue desarrollada por la Universidad de Ciencias Aplicadas de Mittelhessen y a diferencia de las aplicaciones anteriores quería centrarse en que fuera utilizada únicamente por carreras de ramas científicas.

Para entrar en la página y poder usarla no es necesario haber realizado ningún registro previo.

La interfaz inicial muestra:

- Una tabla con los retos creados hasta el momento.
- Un cuadro de texto para crear un nuevo reto o retomar uno ya creado.
- La elección del idioma.
- Elección del tema.

Arsnova permite un modo de juego en el que las preguntas y respuestas son mostradas en el dispositivo del participante. Así como otro, en el que estas sólo se visualizan en la pantalla del profesor y el alumno elige la respuesta en su propio dispositivo.

Una de las opciones a destacar antes de jugar es que da opción a elegir un apodo, pero nunca uno creado por uno mismo, si no cogido de una lista de nombres de personajes famosos, históricos, de película...

En cuanto a la creación del reto, lo primero que nos oferta es poder incluir cualquiera de estas opciones en el reto, pudiendo elegir uno o varios tipos de los indicados:

- Respuesta simple o múltiple.
- Si/No o Verdadero/Falso.
- Respuesta Corta.
- Secuencia.
- Encuestas.

A continuación, se empieza a editar cada pregunta, las cuales no tienen límite de opciones de respuesta pudiendo incluir en ellas emoticonos en lugar de palabras, y como se comentó anteriormente, se puede elegir si mostrar las opciones de respuesta en la pantalla del alumno o no.

El siguiente paso es elegir el tiempo que tienen para responder la pregunta, el tiempo mínimo que ofrece son 10 segundos, pero no hay un máximo establecido.

Una vez finalizado el reto, los datos se guardan en el navegador del profesor, y se le da una opción de descargarlos en Excel si lo desea.

Otras herramientas basadas en retos

Aparte de las herramientas analizadas en las secciones anteriores existen otras con menor impacto en el mercado o con menor grado de madurez, no obstante, se incluyen a continuación al haber sido analizadas como parte del desarrollo de este trabajo:

- Socrative: Los retos son de tipo simple, múltiple y de respuestas cortas, no hay que introducir ningún código para iniciar el juego. También es destacable su versión de pago [11]. Su sitio web es: <https://www.socrative.com>
- Plickers: Se caracteriza porque no es necesario ningún dispositivo, ya sea móvil, tablet u ordenador para conseguir las respuestas, ya que se enfoca en la realidad aumentada. Por eso cada alumno inicialmente recibe un papel con un código y unas pautas de cómo debe ubicarlo dependiendo de la respuesta que quiera dar.

Las respuestas son únicas o múltiples, y en el momento de recopilarlas el profesor apunta con la cámara a la clase, donde estarán sus alumnos con los papeles levantados mostrándoselos. Esta aplicación es muy útil en entornos en los que los alumnos no puedan llevar teléfonos móviles o no haya dispositivos suficientes para todos ellos [12]. Queda alojado en: <https://www.plickers.com>

1.1.3. Resumen comparativo de Quizziz y Kahoot!

En la siguiente tabla se muestran las principales diferencias entre las dos aplicaciones de retos más usadas hasta la fecha, Quizziz y Kahoot!

Quizziz	Kahoot!
Muestra retos más populares	Muestra retos al azar
Se necesitan más usuarios para hacer un reto	Se pueden jugar retos en solitario
No es necesario que el profesor muestre una pantalla mientras se realiza el reto	El profesor debe de estar con una pantalla, mostrando las preguntas y respuestas para que las lea el alumno
Varios idiomas para la aplicación	La página web se encuentra únicamente en inglés
Varios temas para cambiar el diseño	No se puede cambiar el diseño de la página web
Más de una estadística al completar el reto	Solo se puede conseguir la puntuación al finalizar el reto
Preguntas del reto pueden crearse o cogerse de otros retos	Preguntas del reto pueden crearse
Tipos de reto: Simple o múltiple	Tipos de reto: Simple, múltiple, secuencia y sondeo
Tiempo de respuesta: 5 seg - 15 min	Tiempo respuesta: 5 seg - 2 min
Calificar reto por rango de curso	Calificar reto por grupo social
Se puede imprimir el reto creado	No hay opción de descargar el reto creado

1.2. ¿Qué es la gamificación?

La gamificación es un término, que proviene del inglés *gamification*, se refiere al uso de mecánicas y elementos del juego en un ambiente no lúdico, es decir, aprender usando técnicas y objetos del juego [13] [14]. Tiene como fin motivar y hacer participar a un grupo de personas.

Actualmente se está aplicando en distintas áreas como marketing, empresas, docencia, recursos humanos, etc. [14] En este proyecto nos hemos enfocado en la docencia universitaria.

La gamificación enfocada en la docencia es un avance en cuanto a la mejora de concentración, motivación y rendimiento de los alumnos, ya que en la actualidad los videojuegos están muy presentes en el día a día de los estudiantes, siendo una afición muy frecuente en la sociedad actual, y aunque la gamificación no sea un videojuego sí que recoge algunos de sus elementos.

El objetivo de la gamificación aplicada en la educación es que los alumnos se comprometen más con la dinámica de la clase, lo que conlleva un incremento de sus conocimientos y en muchos casos incentiva el trabajo en equipo.

El elemento principal de la gamificación son sus mecánicas de juego, que son las pautas que han de seguir los usuarios. Además, en algunos de los casos puede crear cierta dependencia a ellos, como si de un videojuego se tratase. La gamificación lleva consigo una serie de técnicas y componentes que nos ayudan a entender esta estrategia de docencia, entre los más destacados tenemos las características de la figura que se muestra a continuación:

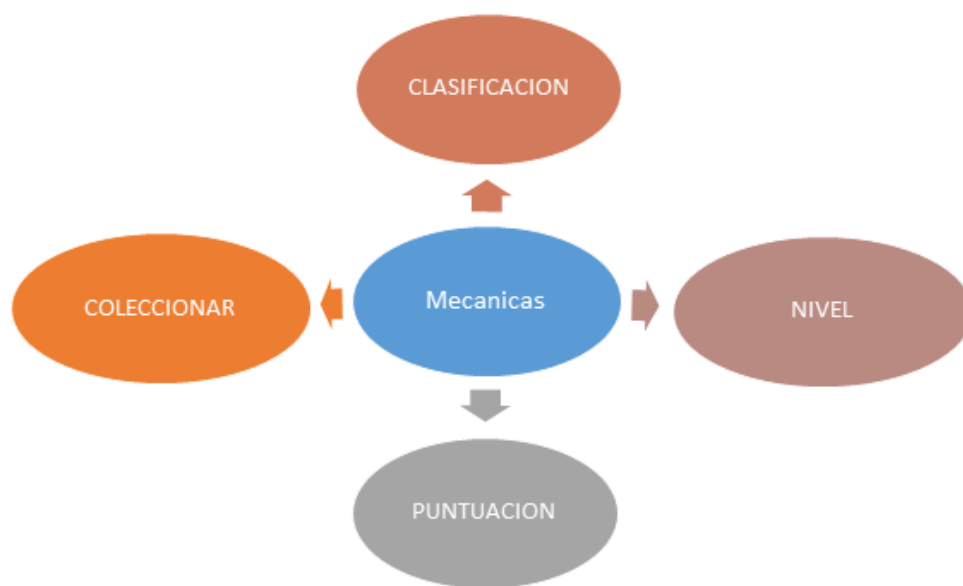


Figura 1. Características de Gamificación [15].

En cuanto a las **mecánicas** de juegos más conocidas dentro de la gamificación tenemos [14]:

- **Coleccionar:** Hace que todos los alumnos quieran conseguir todos los premios o insignias, o el mayor número posible de ellos porque habitualmente su consecución conlleva beneficios.
- **Puntuación:** Cuantos más puntos se obtengan, mejor. Y en este caso mayor nota conseguirá el alumno, en muchos casos ayudará a conseguir premios o insignias.
- **Clasificación:** Ofrece una ordenación por puntuación, tiempo o cualquier otro parámetro. De tal manera que aumenta la competitividad y motivación entre participantes, para intentar alcanzar los mejores puestos de la clase.
- **Nivel:** A parte de conseguir premios e insignias con los puntos, en muchos casos, también se puede subir de nivel, lo que conlleva el aumento de la categoría del participante.

Otro punto muy importante a tener en cuenta en la gamificación son las **dinámicas** de juego, gracias a ellas los alumnos se motivan e implican más en las clases.

Las dinámicas más importantes son [16]:

- **Premios:** Con la obtención de ellos, se consigue que el alumno siga mostrando interés por los nuevos retos.
- **Torneo o Campeonato:** Puede realizarse de forma individual o colectiva.
- **Cooperación:** Ayuda entre los integrantes del mismo grupo.
- **Compañerismo:** Ayudar a la pareja o resto del grupo sin esperar nada a cambio, y en beneficio del equipo.

La gamificación se basa en la psicología positiva y en la teoría de la determinación, así como en el conductismo. Las dos primeras se apoyan en motivaciones intrínsecas del individuo, en el hecho de que el ser humano tiene una necesidad de ser social y competente [13].

El conductismo se basa en motivaciones extrínsecas a él, consiste en integrar logros y recompensas que motiven al alumno, aunque debemos de prestar especial atención a las recompensas porque pueden llegar a provocar el efecto contrario al que buscamos, si la dificultad de conseguirlas es tan grande que produzca que los alumnos se rindan [13].

También debemos tener en cuenta, al aplicar la gamificación, que existen distintos tipos de usuarios, porque no todos se centran en conseguir las mismas metas. Algunos querrán terminar los retos al 100%, consiguiendo todas las recompensas posibles y dándoles igual la puntuación; para otros su única meta es conseguir todos los puntos posibles y no bajar nunca de ese primer

puesto. Y otros usuarios que ya no pueden conseguir ninguna de las metas anteriores, se centran en ayudar a los demás a que lo consigan [13].

Los beneficios que se pueden conseguir con la gamificación son [13]:

- **Compromiso:** Tener retos cada corto periodo de tiempo, hace que los alumnos intenten llevar todo al día, y así conseguir buenos resultados al realizarlo.
- **Motivación:** El alumnado, al tener la posibilidad de conseguir recompensas o buenas puntuaciones en el ranking, se motiva e intenta obtener los mejores resultados de la clase.
- **Conexión social:** Algunos retos, al ser cooperativos, enseñan a los alumnos primordialmente a relacionarse con otras personas y a aprender a trabajar en equipo, ayudándose unos a otros. Esto favorece la unión de la clase y el compañerismo.
- **Competencia:** Los alumnos, al tomárselo en serio, consiguen que todos se vayan esforzando más a medida que transcurre el tiempo. Fomentando así, un clima de competitividad y de compañerismo.



Figura 2. Esquema de Gamificación [13].

En resumen, lo que se intenta con la gamificación es que los alumnos presten más atención al temario, puesto que pueden ser preguntados sobre ello y por tanto se mejora el rendimiento.

Todo mediante el fomento de la participación, la mejora de atención, el tiempo de respuesta y la dinámica de la clase. Además, el profesorado siempre contará con estadísticas que le servirá para conocer donde fallan sus alumnos y por ende dedicar más tiempo a dichos temas.

1.2.1. Motores de gamificación

La aplicación de la gamificación en una herramienta educativa requiere implementar y embeber uno o varios de los mecanismos de gamificación descritos en la sección anterior. No obstante, ya que estas características pueden ser implementadas en diferentes herramientas han surgido iniciativas denominadas *motores/plataformas de gamificación* que simplifican la integración de las mecánicas de gamificación en una herramienta. A continuación, se analizan algunas de los motores de gamificación de código libre que han sido analizados durante el desarrollo de este trabajo.

Sugar Engine

Sugar Engine es un motor de gamificación muy versátil, cuya API REST permite dotar de gamificación a cualquier desarrollo software que así lo requiera. Sus principales características son:

- Para poder ponerlo en funcionamiento se necesita mandar una petición post con el usuario administrador y su contraseña. Lo cual devuelve un Bearer Token o *autorization*, que será necesario almacenar para mandarlo en la cabecera de cada petición.
- Un usuario se puede incluir dentro de un grupo.
- La única opción de puntuación dentro de un ranking es sumarla.
- Al crear un leaderboard o ranking, siempre antes de usar sus variables, hay que inicializarlas, ya que no lo hace el motor por defecto.
- Al crear una insignia o un logro, siempre se tiene que crear junto a ello un premio.
- Se pueden crear alianzas y desafíos.

ActiDoo Gamification Engine

ActiDoo Gamification Engine, en adelante ActiDoo, el cual es un motor de gamificación de código abierto, con las siguientes características:

- Añadir, actualizar y eliminar un usuario.
- Obtener las insignias del usuario.
- Aumentar el progreso del usuario.

No obstante, algunas de las limitaciones identificadas son:

- No hay una diferenciación de los usuarios por grupo, clase o conjuntos.
- Cuando damos de alta o modificamos un usuario, tenemos la opción de insertar datos como la latitud, la longitud o la zona horaria. Sin embargo, no nos pide ningún atributo más propio de un motor de gamificación.
- No dispone de leaderboard o ranking.
- El alta de una insignia es compleja y poco intuitiva.
- No se puede dar de alta una insignia mediante una petición API REST, sino que sólo se puede hacer a través del panel de administrador de ActiDoo.

En definitiva, Sugar Engine es mucho más completo y sofisticado que ActiDoo. A pesar de ello, las instrucciones dadas en su documentación no son muy precisas y muchas de las opciones que dan no funcionan correctamente o hay que hacer diversas pruebas desde la consola del navegador hasta poder dar con la solución.

1.3. Analíticas

Actualmente vivimos en un mundo digital, sería imposible imaginar nuestra vida sin la tecnología. Estamos rodeados de tecnología desde que nos despertamos hasta finalizar nuestro día. De hecho, según un artículo de la revista *muy interesante* “España lidera el ranking mundial tanto por número de smartphones per cápita como por el uso de internet a través de estos dispositivos” [17] y cada búsqueda que realizamos a través de cualquiera de nuestros dispositivos deja un rastro de todo lo realizado, una huella digital.

De todo esto podemos beneficiarnos a nivel educativo, sí el mundo entero ha cambiado es normal que también cambien las aulas, no solo debe haber una pizarra, sillas y mesas cuando hay un sinfín de posibilidades como pizarras digitales, campus virtuales, libros electrónicos, aplicaciones, etc. [18]

Desde hace tiempo distintas disciplinas han ido realizando análisis y recogiendo datos de instituciones educativas, todas ellas tienen elementos en común como los beneficios, las categorías o los actores [19].

Según Inteligencia de Negocio [20], las diferentes disciplinas de análisis de datos educativos llevan un proceso basado en los siguientes puntos [19]:

- **Captura:** Recogida de datos.
- **Informe y análisis:** Este paso se puede dar en dos pasos distintos.
- **Predecir:** Después de ser analizados varias veces los datos y observar las pautas, se intentarán predecir los futuros resultados.
- **Actuar:** Una vez predichos los datos, aquí es donde se actuaría solucionando los problemas que puedan haberse detectado.
- **Refinar:** Todos los procesos de análisis necesitan ser reajustados siempre que se añada nueva información.

Con ese proceso todas las personas dentro del mundo académico saldrán beneficiadas, aunque como todo lo que conlleva manejo de datos lleva consigo también una serie de problemas que tendrán que irse resolviendo. Siendo el principal el planteamiento de la legalidad y eticidad de la “vigilancia” continua realizada sobre el alumno o estudiante. Los diferentes tipos de disciplinas son [19]:

- **Analíticas académicas:** Esta disciplina fue una de las primeras y su propósito era entender y progresar en cuanto a educación. Se basaba en la información que se almacenaban en los sistemas de información de las instituciones académicas.
- **Minería de datos educativos:** Difiere de la anterior porque recoge datos muchos más directos de la formación de los estudiantes. El fin de esta es encontrar métodos para conocer los procesos educativos y cómo se desarrollan. Esta disciplina también aspira a automatizar procesos de descubrimiento de patrones y centrarse en procesos de aprendizaje concretos [21].
- **Analíticas de aprendizaje:** Disciplina que nació poco después de la Minería de datos educativa compartiendo mismos objetivos, pero también guarda algunas diferencias como, por ejemplo, que investiga nuevos patrones de conocimiento, dejando su interpretación y uso a interpretación humana. Se focaliza más en la conjetura de resultados y en sistemas de alarma [21].

Estas tres disciplinas, aunque son distintas, se encuentran siempre superpuestas.

Las analíticas han sido utilizadas en diferentes campos, cómo [19]:

- **Educación tradicional:** Los primeros estudios acerca de análisis de aprendizaje fueron, en la educación tradicional, mediante la recogida de datos. En la mayor parte de los estudios se recolectan primordialmente datos académicos y demográficos. Estos se dan principalmente en el entorno universitario.

Hay muchos ejemplos de estudios en educación tradicional, uno de los primeros es el de Fausett y Elwasif [22] en el que analizaron las respuestas de los alumnos en los exámenes con el fin de predecir la calificación final que tendrán los mismos.

En general los estudios intentan comprobar la relación que hay entre la calificación en los exámenes para la admisión de la universidad, y cómo le irá al alumno en esa universidad, si podrán finalizar sus estudios y predecir cuáles serán sus calificaciones en un futuro. Es decir, prever la probabilidad de éxito académico de un alumno a partir de sus datos académicos presentes o pasados.

- **Tutores inteligentes:** Se trata de un sistema educativo que pretende simular un tutor humano [23], adaptándose a la evolución del alumnado. En su mayoría son creados para tratar temas concretos y tienen los mismos componentes [24]: módulo experto, módulo del estudiante, módulo tutor e interfaz de usuario. Como el sistema está en continuo contacto con el alumno permite recoger datos del proceso de aprendizaje.

Existen varios estudios que muestran el funcionamiento y los beneficios de los tutores inteligentes, entre ellos se encuentran, por ejemplo, el de Chang et al., en él se analiza si estos tutores facilitan ayuda [25]. Otro importante estudio es el de Yudelson quien consigue clasificar a los alumnos según patrones de aprendizaje [26]. Incluso se han llevado a cabo estudios como el de Hämäläinen & Vinni, que intentan mejorar a estos tutores inteligentes [27].

- **Sistemas de Gestión de Aprendizaje:** Fue un gran avance para los organismos académicos, porque con ello se podía dotar al alumno de manera on-line de apuntes, creación de foros para resolver problemas o dudas, permitiendo de este modo que el alumno y el profesor estén en contacto.

De la misma manera, la información recogida difiere completamente a las anteriores, puesto que se centra en la utilización del alumno de estas plataformas.

- **Analíticas web:** Estudian los accesos de los usuarios a una página web [28], a qué apartados de esta página le dedican más tiempo y las interacciones que realiza. Todo esto tiene importancia principalmente dentro del mundo del comercio, para ver las tendencias del comprador, pero se puede usar de igual modo en la educación y en el aprendizaje.

- **Interacciones sociales:** Los foros de discusión se emplean con bastante frecuencia, en ellos alumnos y profesores pueden comunicarse, exponer ideas, realizar cuestiones, etc., por escrito de forma virtual.

También sobre estos foros hay diversos estudios basándose en las teorías de análisis de redes sociales para saber cuánto se usan, en qué horario, qué alumnos los usan más y quiénes menos, de modo que lo sepa el profesor y pueda actuar según su criterio [29]. Asimismo, hay estudios que intentan predecir el resultado académico que tendrán los alumnos analizando las participaciones en los foros [30].

- **Análisis de texto:** Trata de analizar los comentarios que los alumnos realizan en los foros con varios objetivos, estos serían computar la participación, analizar el proceso de los debates, obtener conclusiones de la productividad del alumnado y encontrar en los debates preguntas a las que no se hayan podido dar respuesta [31] [32].

- **Cuestionarios:** Los cuestionarios usan la forma más clásica de evaluar a alumnos, pero aportan una serie de beneficios con respecto a los test convencionales, tanto para el alumnado como para el profesorado. La autocorrección ahorra tiempo al profesor que podrá invertirlo en otras tareas y al alumno le sirve como herramienta para autoevaluarse y poder conocer sus fallos.

Hay diversos estudios sobre esta herramienta, algunos demuestran su parte negativa, como Hernández [33] quien avisa de las posibilidades de copiar, hacer trampas y de falsear resultados. Para solucionar estos posibles fraudes por parte de los alumnos, Hernández, Ochoa, Muñoz y Burlaka [33] en 2006 proponen un método que analiza las veces que han entrado a realizar el examen y los patrones de respuestas, descubriendo así quienes hacen trampa.

- **Cursos on-line:** Como su nombre indica, son cursos que pueden ser realizados por cualquier persona, ya que son a través de vídeos de forma on-line, los cuáles también adjuntan otros métodos de estudio para su ayuda.

Una de las grandes ventajas que tienen estos cursos respecto a las analíticas es la gran cantidad de personas que lo utilizan ya que, al tener un ámbito mundial, hay gente con todo tipo de características. Aunque el mayor problema es que un gran número de ellos no llegan a finalizarlo.

1.3.1. Herramientas de analíticas de aprendizaje

Las herramientas usadas para las analíticas de aprendizaje suelen ser creadas para trabajar en medios concretos. Actualmente, la lista de herramientas de la que disponen los profesores es bastante reducida, pero aun así se pueden destacar los siguientes tres tipos de categorías [34]:

- **De propósito general aplicadas a la educación:** Por ejemplo, como el uso de google analytics.
- **Embebidas dentro de la aplicación:** Por ejemplo, moodle incluye un módulo de analíticas.
- **Plataformas de analíticas:** Cubren alguno o todos los aspectos principales de las analíticas de aprendizaje.

2. Objetivos de este trabajo

Las herramientas como Kahoot! y Quizizz pese a ser fáciles de utilizar y tener una gran acogida, sin embargo son herramientas se utilizan de manera aislada, es decir, se encuentran desconectadas del resto del plataformas o herramientas. Más aún, como parte del proceso de aprendizaje es posible que el profesor cree más de un reto para el mismo conjunto de alumnos, pero estas herramientas no permiten gestionar la experiencia de aprendizaje como el conjunto de retos relacionados (forman parte de la misma experiencia educativa o del mismo curso / asignatura) sino que son actividades independientes. Por otro lado, las funcionalidades de gamificación y analíticas también vienen prefijadas y son limitadas, por ejemplo, en el caso de las analíticas éstas se limitan a un informe final tras terminar el reto donde no todas las interacciones del alumno pueden quedar reflejadas.

El objetivo general de este trabajo consiste en la creación de una plataforma integrada que permita a un profesor crear nuevos retos para sus alumnos (organizados en grupos y asignaturas), definir elementos de gamificación para sus alumnos y recopilar eventos más relevantes como resultado de la interacción de los alumnos con los retos. La plataforma estará compuesta por los componentes que se muestran en la Figura 3.

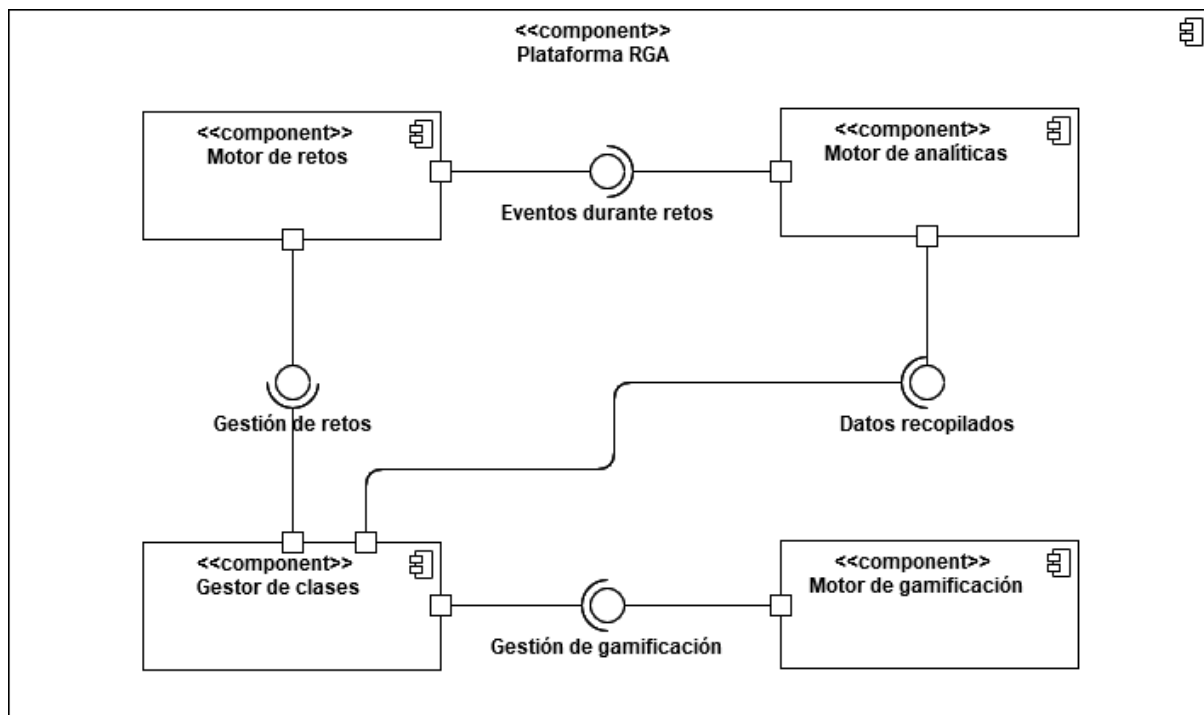


Figura 3. Diagrama de componentes de la Plataforma RGA.

Asimismo, como objetivo secundario se pretende definir el conjunto de interfaces / APIs necesarias para que los diferentes componentes puedan sustituirse por otros.

Más concretamente, el objetivo de cada componente del sistema es:

- **Gestor de clases:** Este módulo se encarga de aglutinar y proporcionar una experiencia unificada del sistema, cuya función específica consisten en gestionar usuarios, clases y grupos de alumnos y profesores.
- **Motor de retos:** El motor de retos es el encargado de generar retos compuestos por diferentes preguntas. A su vez, sobre este módulo se llevará a cabo la resolución de los mencionados retos por parte del alumnado.
- **Motor de gamificación:** El módulo de gamificación tiene como objetivo dotar del concepto de gamificación, expuesto en la sección [1.2. ¿Qué es la gamificación?](#), a los diferentes retos que hayan sido completados. Como metas principales de este módulo tenemos:
 - Generación de insignias.
 - Incremento de puntuaciones.
 - Clasificación por leaderboards o rankings.

- **Motor de analíticas:** En lo que se refiere al módulo de analíticas, la aplicación será capaz de generar una serie de registros en los que aparezcan las diferentes respuestas de los estudiantes y si son correctas o no, además del tiempo que se ha empleado en responderlas. Con el objetivo de poder procesar, visualizar y estudiar esta información.

En definitiva, con todo ello se pretende conseguir:

- Crear una aplicación que integre retos, analíticas y un motor de gamificación, es decir, una aplicación en la que se empleen técnicas y dinámicas características del juego y el ocio en actividades no lúdicas, en este caso en docencia.
- Dotar de una herramienta de trabajo para el profesorado, que facilite el seguimiento del alumnado mediante estadísticas.
- Mejorar el rendimiento de los alumnos que aprovechen esta plataforma, gracias a todos los beneficios de la gamificación.
- Fomentar la competitividad, participación y motivación de los estudiantes que hagan uso de esta aplicación.
- Incrementar el interés por el estudio de los mencionados alumnos.
- Poder examinar los conocimientos del estudiantado de forma sencilla y bajo el soporte de un entorno digital.

3. Plan de trabajo

3.1. Metodología de desarrollo

En cuanto a la Metodología de Desarrollo se refiere, hemos hecho uso de una metodología ágil combinando diferentes características entre Kanban y Scrum.

Los aspectos utilizados de Scrum han sido:

- Reuniones periódicas, aunque no diarias, un par de veces a la semana.
- Reunión de planificación al comienzo del desarrollo del proyecto.
- Visualización del trabajo, elemento que comparte con Kanban, el cual queda explicado más adelante, en sus características.

Las características extraídas de Kanban son:

- Visualización del trabajo, con un tablero de Trello, dividido en diferentes estados por los que van transcurriendo las tareas. En nuestro caso, estos estados han sido: Pendiente, En proceso, Bloqueado, Pruebas y Hecho.
- Sin roles definidos.
- Entregas continuas.
- Límite del Work In Progress: Dos tareas como máximo por cada miembro del equipo.
- Sin estimación de tareas.
- Sin priorización de tareas.

Por otro lado, de Scrum no hemos implementado ni reunión de revisión, ni tampoco una reunión de retrospectiva. No hemos hecho uso de Sprints, ya que la entrega ha sido continua e incremental. Del mismo modo que tampoco hemos hecho uso de los roles de Scrum, aunque se puede presuponer que el papel de Product Owner lo ha tenido nuestro director de proyecto, Iván Martínez Ortiz, y como equipo de desarrollo seríamos Carlos Marrón Díaz y Jorge Merino Martínez. Aun así el papel de Scrum Master, en este caso, ha sido inexistente.

En cuanto a Kanban no hemos hecho uso del Lead Time, ni del Cycle Time.

3.2. Planificación del proyecto

En lo que respecta a la planificación del proyecto, se estableció que la metodología que se iba a utilizar sería ágil. Definiendo así los puntos principales que se mencionan en el apartado anterior, [3.1. Metodología de desarrollo](#).

De igual modo definimos la arquitectura y el modelo, los cuales se fueron mejorando a medida que avanzaba el proyecto. Ver apartado [4.2. Arquitectura del software de Plataforma RGA](#).

Uno de los elementos fundamentales en una combinación de metodologías entre Scrum y Kanban, es la Visualización de las tareas y del trabajo. Para lo cual, entre diferentes herramientas como Jira o Trello, se ha elegido esta última ya que es la que más cómoda ha resultado y la que mejor se ha adaptado a las necesidades del proyecto. Estos han sido algunos de los ejemplos de los proyectos Plataforma RGA y Quiz. Ver figuras 4 y 5, respectivamente.

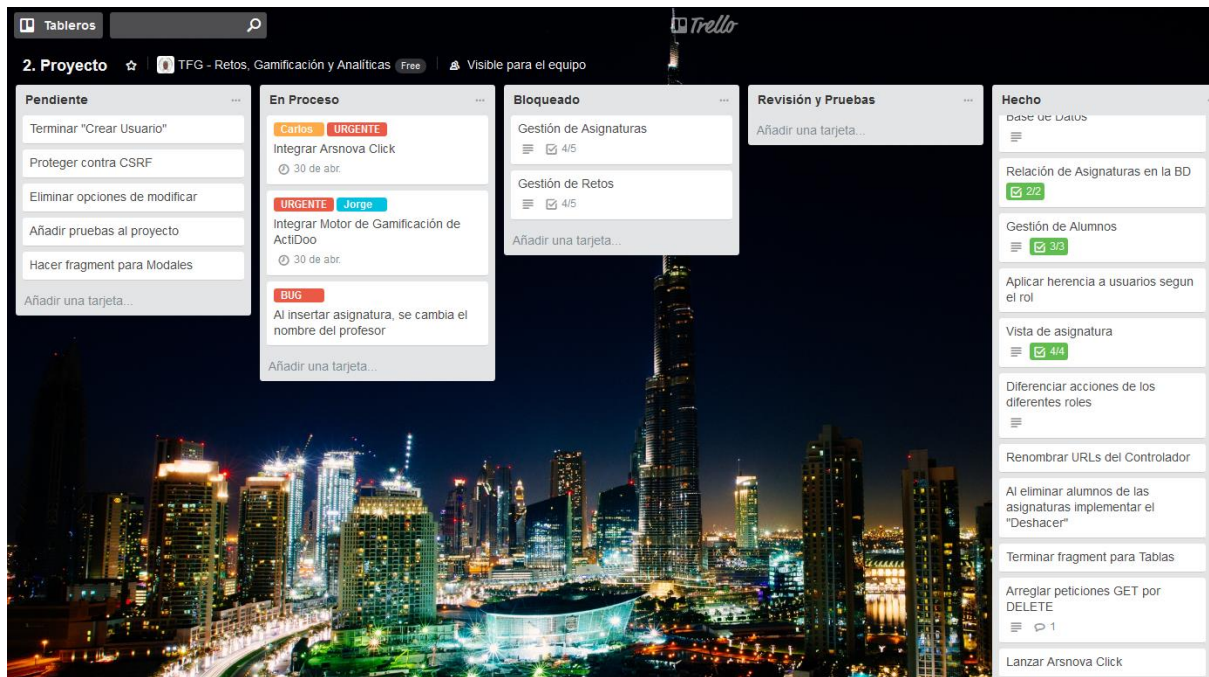


Figura 4. Ejemplo de planificación de Plataforma RGA.



Figura 5. Ejemplo de planificación de Quiz.

4. Resultados

4.1. Evolución del desarrollo del proyecto y descripción de la solución final integrada

4.1.1. Motor de retos

Como se ha descrito en el capítulo [2. Objetivos de este trabajo](#), el objetivo secundario del proyecto ha sido crear una arquitectura modular en base a definir interfaces y APIs para permitir que los principales componentes de la aplicación pudieran reemplazarse, más concretamente, el motor de retos, el motor de gamificación y el motor de analíticas.

Con esta idea, el proyecto comenzó analizando la viabilidad de integrar Kahoot! o Quizizz como herramienta de retos, sin embargo, ninguna de las dos proporciona una API pública por lo que el proceso requeriría o bien realizar ingeniería inversa al cliente web o realizar *screen scraping* sobre la aplicación. Ambas opciones fueron descartas debido a la complejidad, fragilidad y a los aspectos legales de su aplicación.

Descartadas las opciones comerciales, analizamos el uso de arsnova.click como herramienta de retos debido a que proporciona un conjunto de funcionalidades similares (pero limitadas) a Kahoot! y ser una herramienta open source. Arsnova.click, como decisión de diseño, no guarda ninguna información del lado del servidor y, en el caso de los resultados de los retos, se pierden una vez el profesor termina el reto. Por tanto, Tras lograr crear una instalación local arsnova.click y realizar algunas pruebas de concepto, evaluamos la viabilidad modificar la herramienta para proporcionar las funcionalidades necesarias para el proyecto. Las principales conclusiones de este análisis fueron: 1) arsnova.click proporciona una escasa documentación de desarrollo disponible del proyecto, 2) las modificaciones requerirían aprender todo un conjunto de tecnologías totalmente diferentes a las seleccionadas para el resto del proyecto y 3) la tecnología base de arsnova.click (Meteor) no es una de las tecnologías con gran aceptación parte de la comunidad de desarrolladores, por lo que los recursos, ejemplos y herramientas son limitados comparados con otras tecnologías. Como resultado de esta evaluación se descartó arsnova.click debido a la incertidumbre del esfuerzo necesario para realizar las modificaciones sin tener en cuenta otros posibles problemas en el desarrollo.

Como resultado, se decidió implementar una herramienta, denominada Quiz que, pese tener una funcionalidad limitada comparada a Kahoot!, Quizizz o arsnova.click, permitiera poner en práctica la solución completa y definir las interfaces necesarias para que en un trabajo futuro se pudiera abordar la integración con otros motores de retos.

4.1.2. Motor de gamificación

Como parte del desarrollo se analizó desde el punto de vista técnico las dos herramientas descritas en el apartado [1.2.1. Motores de gamificación](#): ActiDoo y Sugar Engine.

En una primera fase se realizó una prueba de concepto con ActiDoo y pese a tener éxito, con el proyecto más avanzado al intentar implementar otras funcionalidades, estas no eran lo suficientemente completas y en algunos casos ni tan siquiera estaban provistas por el motor de gamificación. En la siguiente figura se mostrarán las peticiones disponibles.

```
Get Progress  
GET to "/progress/{userId}"  
  
Increase Value  
POST to "/increase_value/{variable_name}/{userId}"  
  
Add or Update User  
POST to "/add_or_update_user/{user_id}"  
  
Delete User  
POST to "/delete_user/{user_id}"  
  
Get Achievement Level  
GET to "/achievement/{achievement_id}/level/{level}"  
  
Register Device  
POST to "/register_device/{user_id}"
```

Figura 6. Peticiones REST de ActiDoo.

Como se puede observar es un motor que carece de abundantes opciones, e incluso, algunas de ellas no son relevantes para la gamificación. De todas ellas se implementaron las siguientes:

- *Get Progress*: Para obtener los logros de un usuario.
- *Increase Value*: Para aumentar la puntuación de un usuario.
- *Add or Update User*: Para dar de alta o actualizar un usuario.

Dado que no fue suficiente para cubrir las necesidades de la Plataforma de Retos, Gamificación y Analíticas que se estaba implementando, se decidió reemplazar por el motor de gamificación de Sugar Engine, descartando por tanto, el uso de ActiDoo.

4.1.3. Analíticas

Debido a los problemas encontrados durante el desarrollo de los otros dos módulos del proyecto ha impedido el desarrollo ideado para el módulo de analíticas, en particular, la integración con una la plataforma de analíticas como RAGE Analytics (<https://github.com/e-ucm/rage-analytics/>).

No obstante, se ha implementado una funcionalidad básica de registro de eventos relevantes en Quiz y que están accesibles desde la plataforma RGA. De este modo, se ha abordado de manera básica, la recopilación de información de la herramienta generada durante el reto. Junto a ello se han desarrollado, con el fin de complementar y reforzar este módulo, una serie de gráficos analíticos en los que poder visualizar el progreso, las respuestas, el tiempo y los aciertos de los alumnos, así como cierta información derivada de las preguntas de un mismo reto. Todo ello puede ser consultado a través de la solución integrada.

4.1.4. Solución final integrada

Teniendo en cuenta la evolución del proyecto descrito en las secciones anteriores, la solución final está compuesta por tres aplicaciones independientes:

- **Quiz:** De manera resumida, Quiz es un motor de retos / tests que permite dar de alta retos por parte del profesor, así como incluir preguntas con sus diferentes opciones y ajustar el tiempo máximo del que dispondrán los alumnos para responder dicha pregunta. De modo que, una vez el reto haya sido creado, el profesor pueda lanzarlo para que sus alumnos lo resuelvan. Además de ello, Quiz es capaz de almacenar la información relativa a las respuestas del alumnado, con el fin de poder ser exportada para su posterior estudio y análisis.
- **Sugar Engine:** El cual queda detallado en el apartado [1.2.1. Motores de gamificación](#). Para su instalación local se ha hecho uso de Vagrant y Docker en un entorno virtual de Linux, explicado en el [Anexo I: Manual de instalación](#).

- **Plataforma Integrada:** Este es el punto de entrada a la solución completa permitiendo dar de alta alumnos, profesores y asignaturas. Además de ello integra Quiz, lo que permite dar de alta retos, para posteriormente ser gamificados por el motor de gamificación y analizados por el módulo interno de analíticas que contiene la Plataforma RGA.

El diagrama de componentes de la solución final sería el que se muestra en la figura, a continuación.

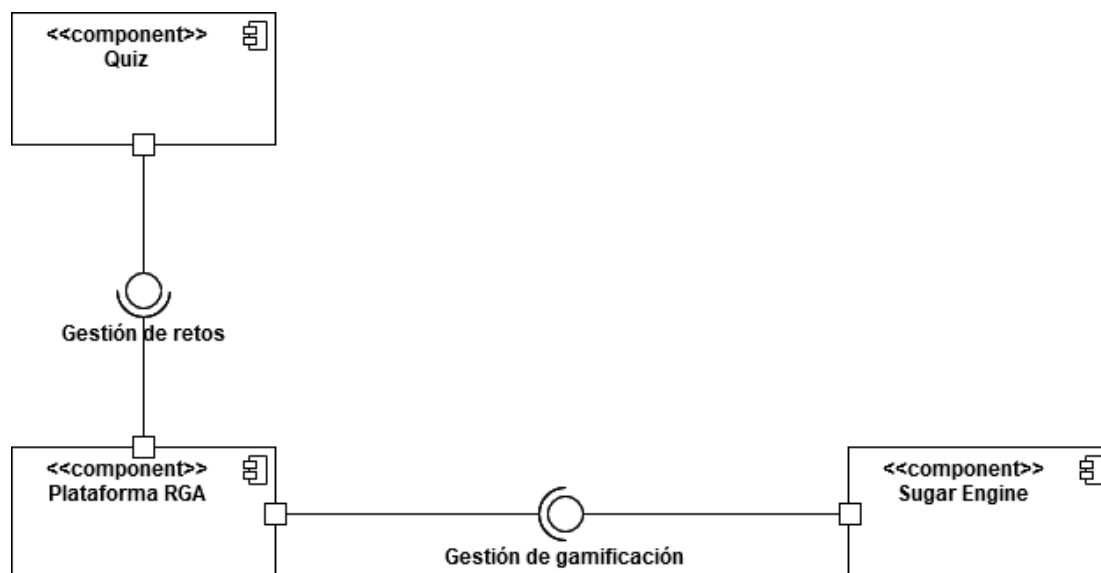


Figura 7. Diagrama de componentes de la solución final integrada.

4.2. Arquitectura del software de Plataforma RGA

La arquitectura que se ha escogido para la Plataforma RGA, ha sido una arquitectura multicapa, basada en los siguientes niveles:

- **Capa de presentación:** Aquella relativa a la gestión de vistas y a la visualización de la interfaz. Cabe mencionar que, debido a la plataforma tecnológica elegida, Spring Boot, se está utilizando el patrón MVC.
- **Capa de negocio:** Es la capa que contiene toda la lógica y reglas de negocio.
- **Capa de integración:** También conocida como capa de datos, es la que se centra en el correcto almacenamiento de los datos en la base de datos. Esta capa se ha reducido y simplificado gracias al uso de JPA, que es el encargado de implementar el Patrón Almacén del Dominio.

El motivo principal del uso de una arquitectura multicapa ha sido el de desacoplar, con la finalidad de hacer mucho más modular la estructura software de la aplicación.



Figura 8. Esquema de la arquitectura multicapa.

4.2.1. Modelo del dominio

El modelo del dominio establecido para la aplicación base ha sido el que se muestra en la figura 9. Con un total de cinco entidades tal y como se describe a continuación.

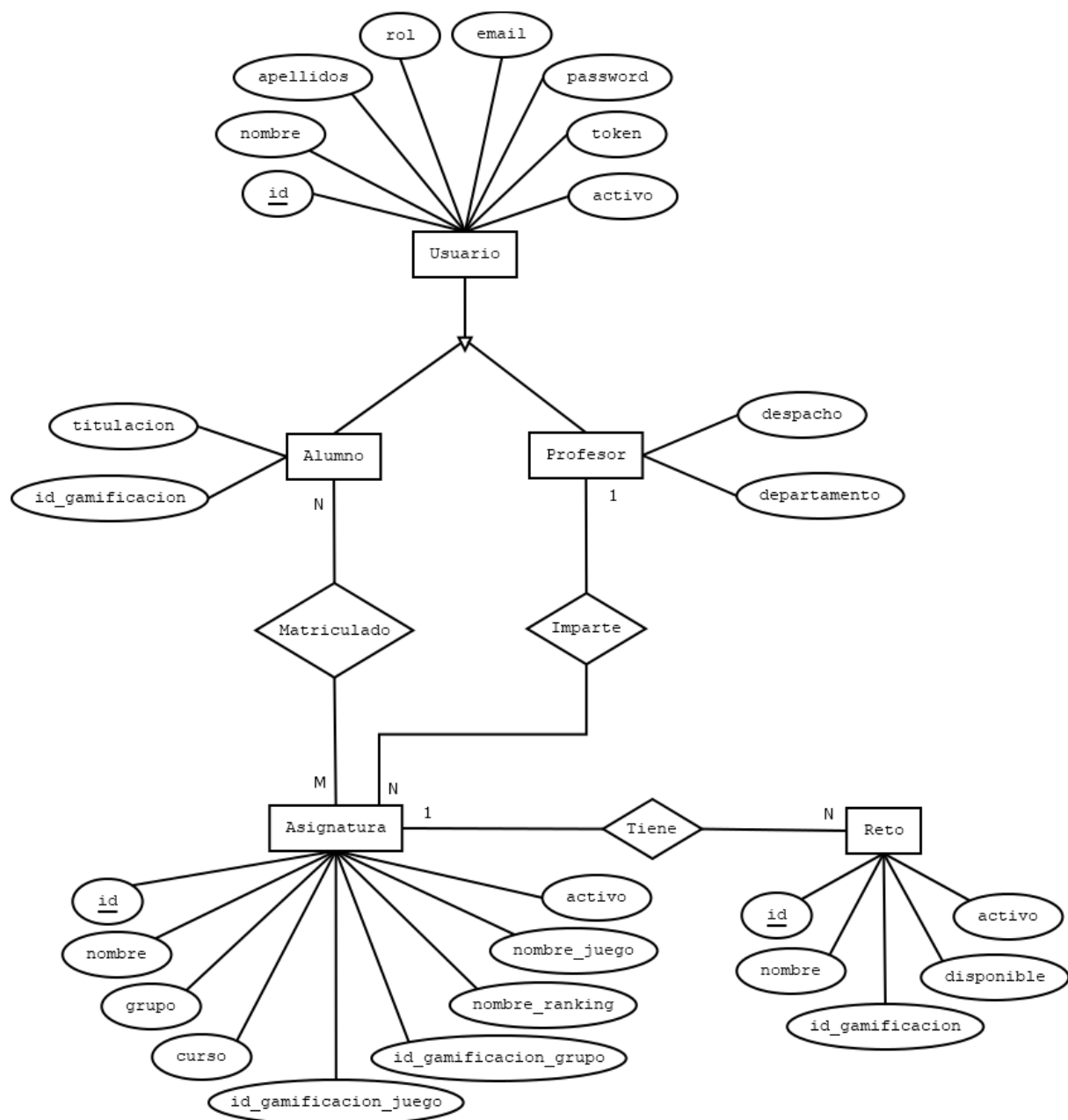


Figura 9. Modelo del dominio de Plataforma RGA.

4.2.2. Diagrama de clases de negocio

Para el anterior modelo del dominio se ha diseñado la siguiente arquitectura software de negocio, representada con el diagrama de clases UML que se muestra en la siguiente figura. Además de contener las Entidades definidas en el modelo del dominio, también está compuesto por los diferentes Servicios de Aplicación, que contendrán toda la lógica de negocio.

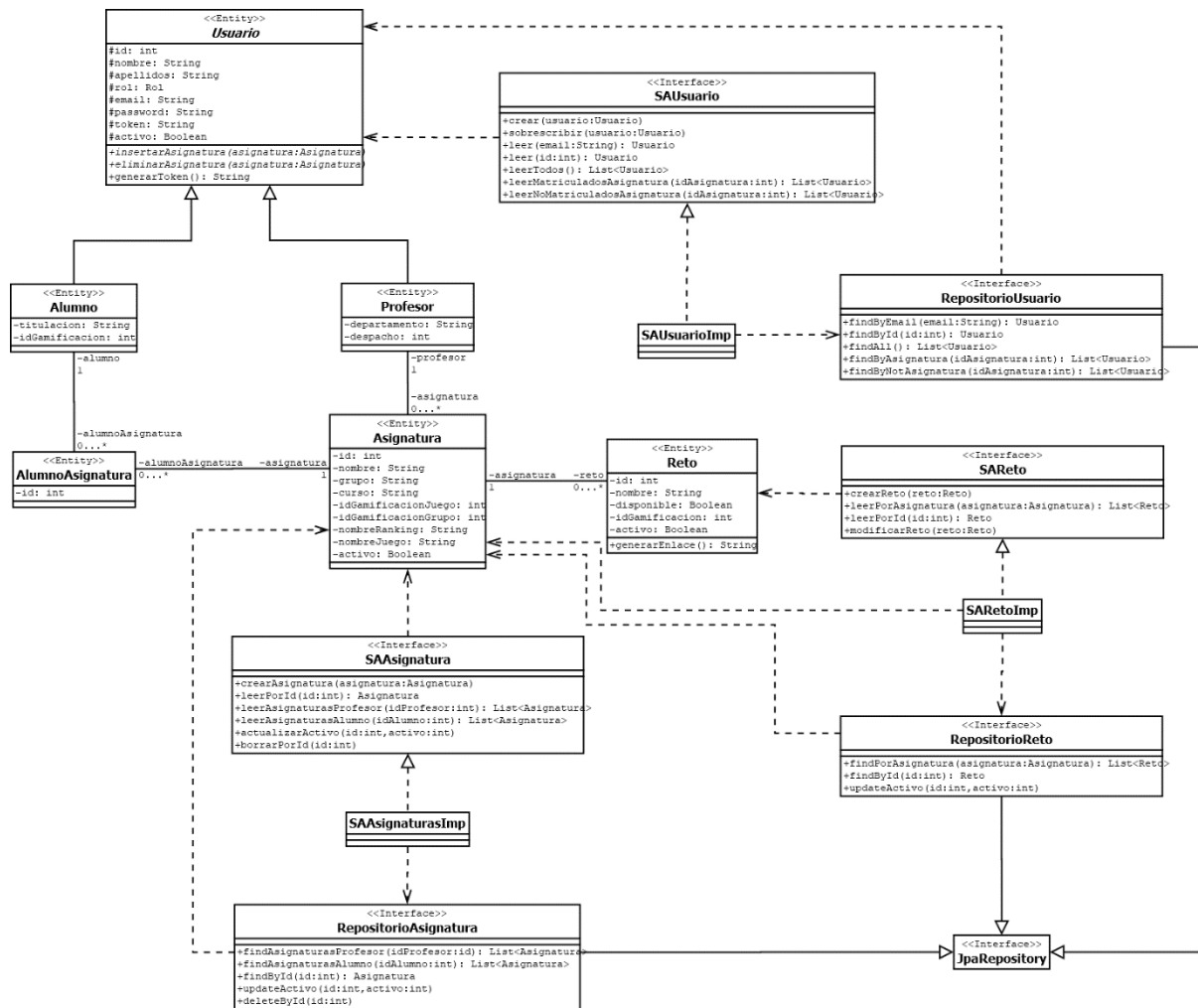


Figura 10. Diagrama de clases de negocio de Plataforma RGA.

4.2.3. Diagrama de clases de presentación

A continuación, se detallarán los diagramas de clases de presentación diseñados para la Plataforma RGA. En ellos se detallan los diferentes métodos del Controlador, así como su dependencia con ModelAndView, ver figura 11.

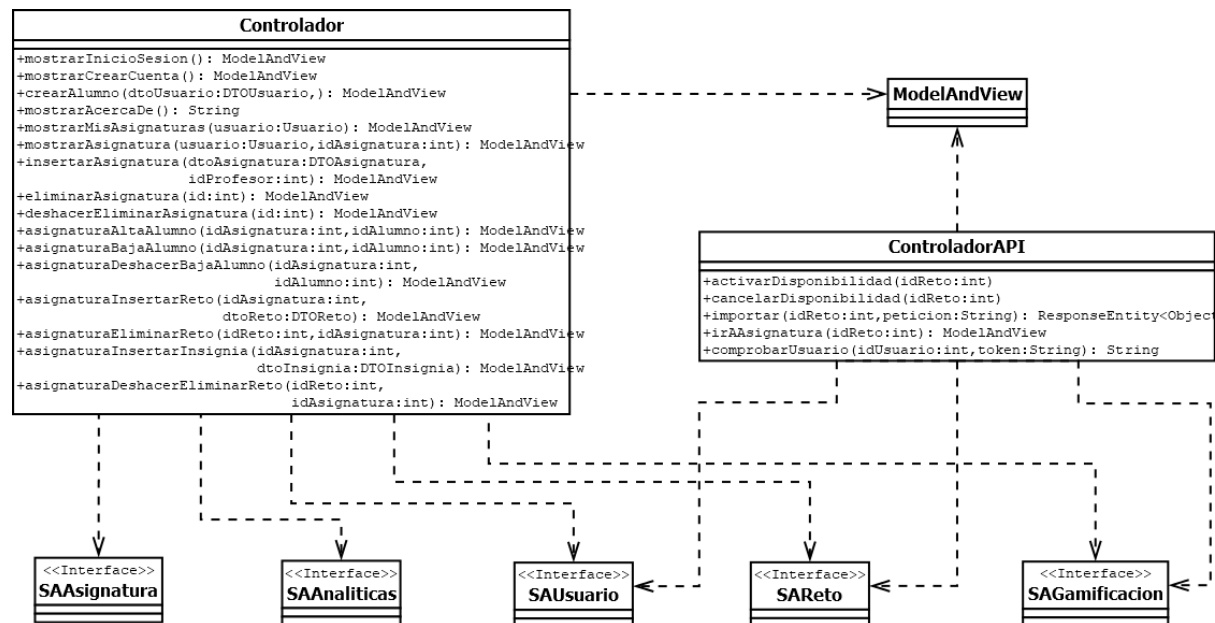


Figura 11. Diagrama de clases de presentación de Plataforma RGA (1).

Por otro lado, en la siguiente figura se muestran los Servicios de Aplicación de los que hace uso el Controlador, diferenciados por colores dependiendo del módulo del proyecto al que pertenezcan. En azul, la Gestión de Clases; en rojo, el Gestor de Retos; en naranja, el Gestor de Gamificación y en verde, el Gestor de Analíticas.

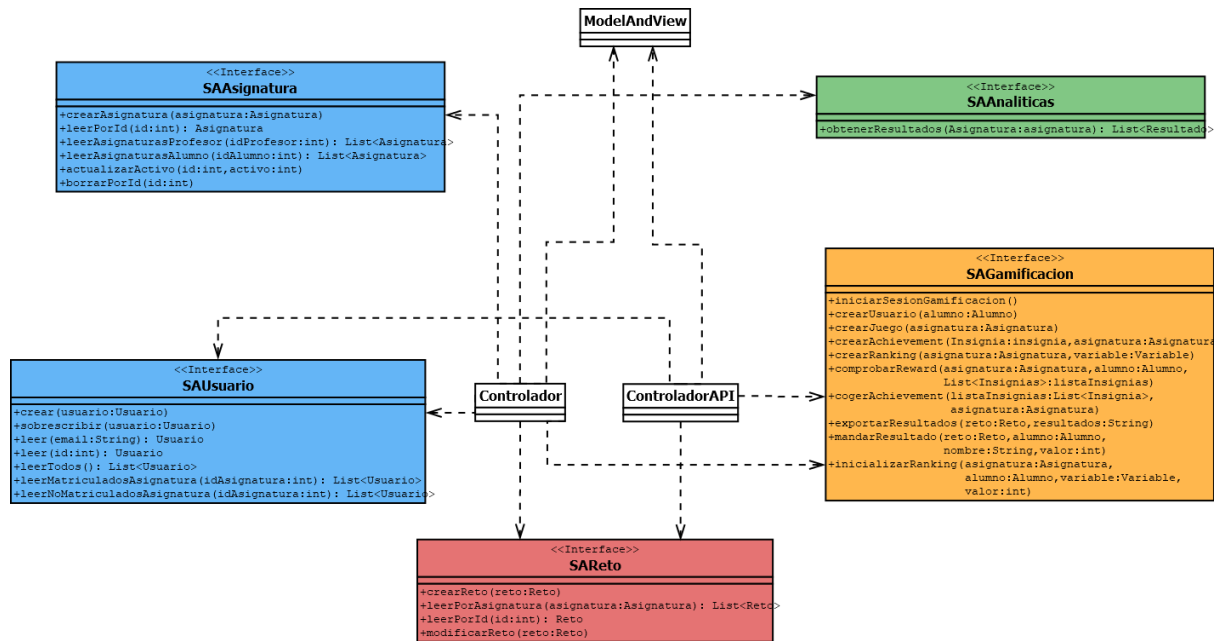


Figura 12. Diagrama de clases de presentación de Plataforma RGA (2).

4.3. Arquitectura del software de Quiz

De igual modo que la arquitectura del software de la Plataforma RGA, ver punto [4.2. Arquitectura del software de Plataforma RGA](#). La arquitectura software para Quiz también ha sido de tres capas, o multicapa, compuesta por presentación, negocio e integración. Y de la misma manera se ha implementado el Patrón Almacén del Dominio con JPA.

4.3.1. Modelo del dominio

En el caso de la aplicación Quiz el modelo del dominio ha sido el establecido en la figura 13.

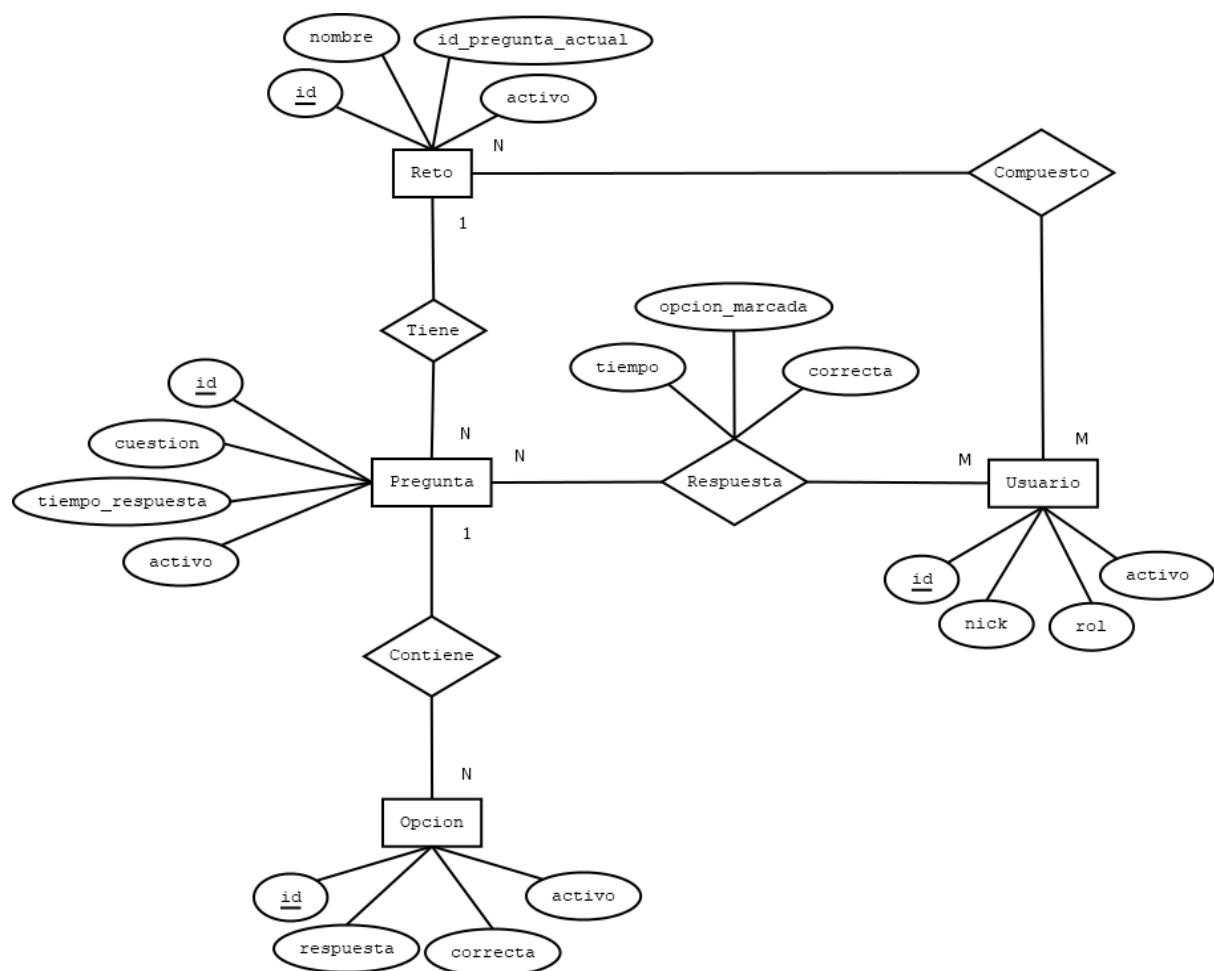


Figura 13. Modelo del dominio de Quiz.

4.3.2. Diagrama de clases de negocio

Siendo su correspondiente diagrama de arquitectura UML, el que se muestra a continuación. En el que quedan fijados los Servicios de Aplicación, con la lógica de negocio y la interfaz de los Repositorios, encargada de acceder a la base de datos.

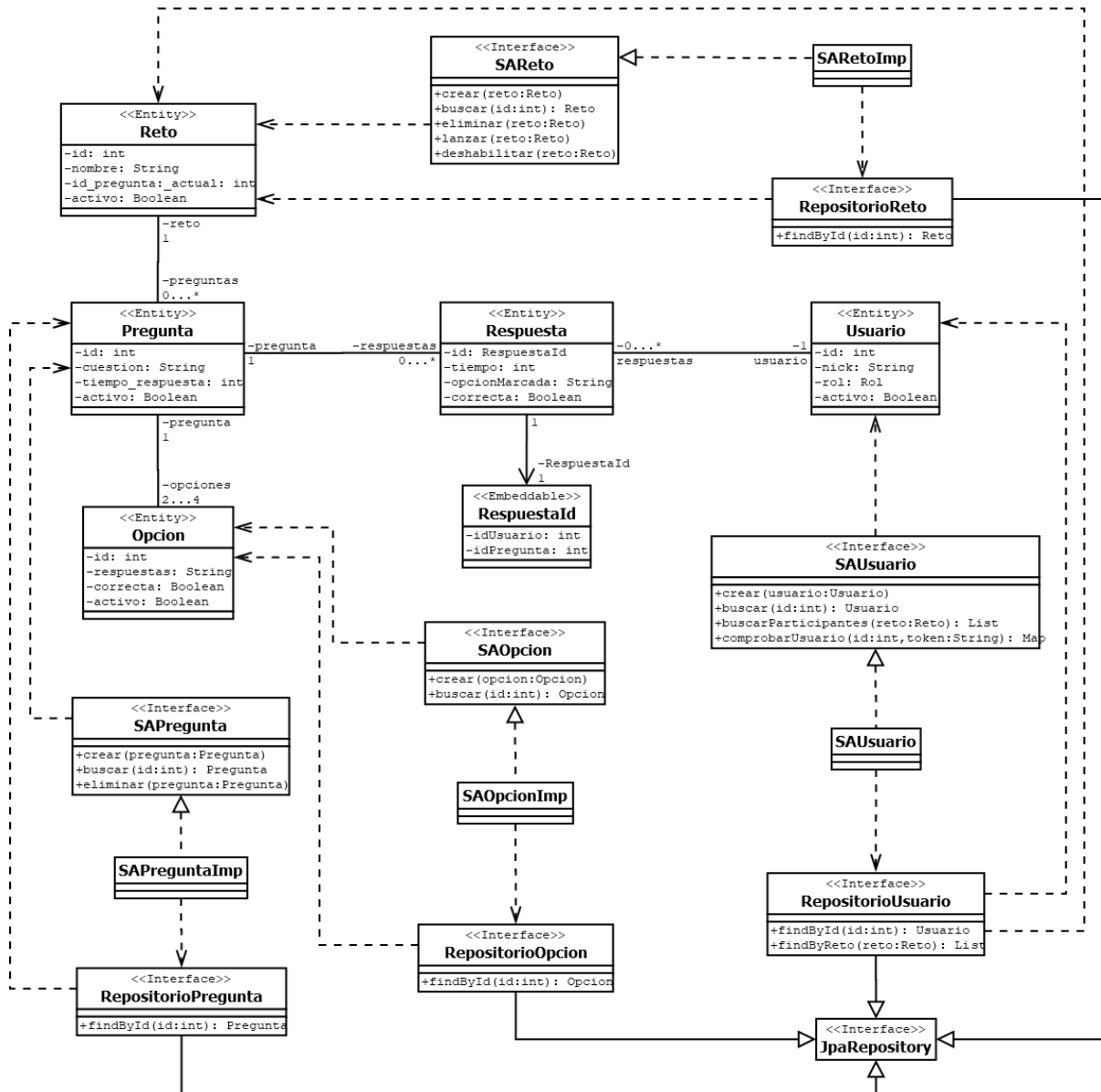


Figura 14. Diagrama de clases de negocio de Quiz.

4.3.3. Diagrama de clases de presentación

En las siguientes figuras se detallarán las diferentes clases que componen el diagrama de presentación, entre los que destacan ModelAndView, el Controlador y sus dependencias con las interfaces de los Servicios de Aplicación.

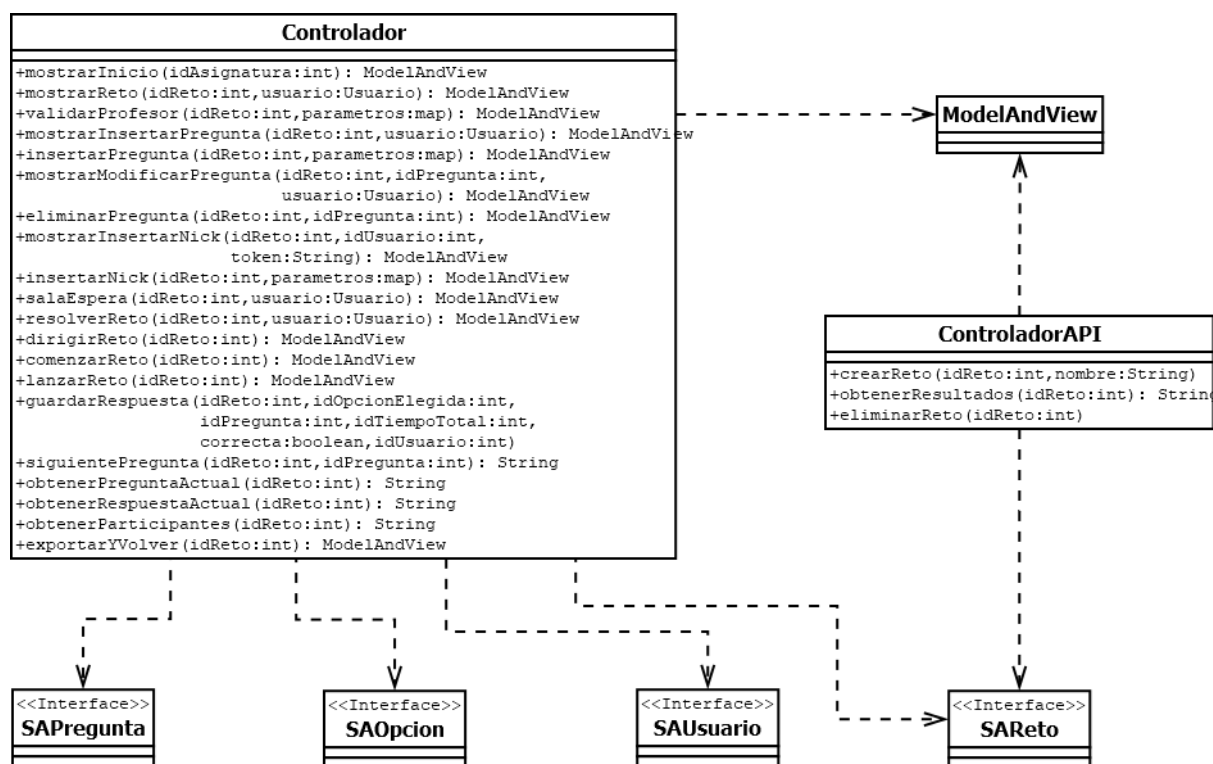


Figura 15. Diagrama de clases de presentación de Quiz (1).

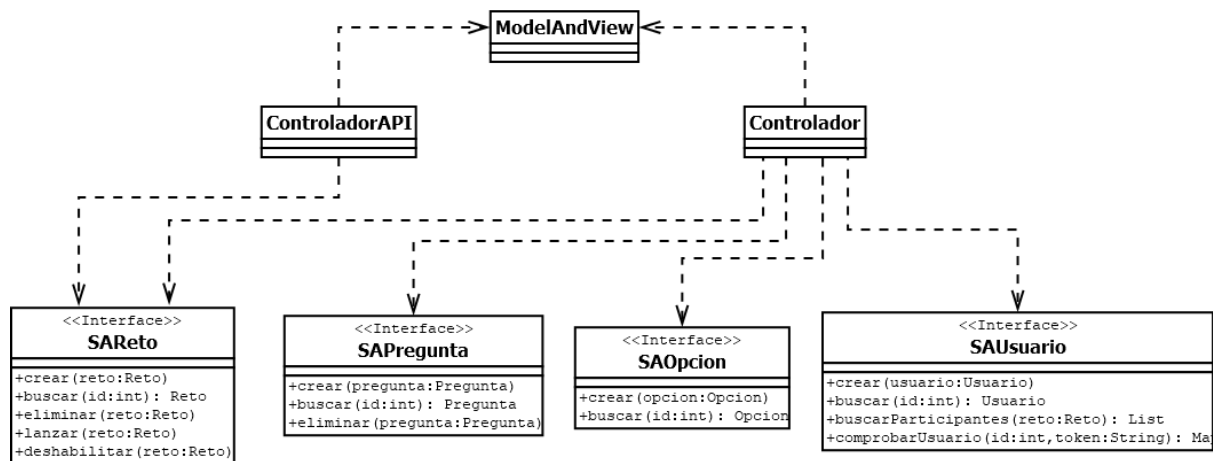


Figura 16. Diagrama de clases de presentación de Quiz (2).

4.4. Tecnologías utilizadas

Para el desarrollo de las aplicaciones hemos utilizados las siguientes herramientas y tecnologías.

Para el front-end:

- **HTML5:** Lenguaje de Marcado de Hipertexto, utilizado para la visualización de las vistas web, e implementado mediante plantillas.
- **CSS3:** Lenguaje de estilos mediante el cual se define y diseña la forma en que se muestran las diferentes páginas de nuestras aplicaciones web.
- **jQuery:** Librería de JavaScript, mediante la cual hemos definido los diferentes eventos y efectos dinámicos, en las vistas de las aplicaciones. En un principio, dicha implementación se hizo con JavaScript, sin embargo, dadas las necesidades del proyecto decidimos finalmente utilizar JQuery. Una de estas necesidades fue la de requerir una serie de peticiones asíncronas y en segundo plano, para lo cual utilizamos AJAX.
- **Bootstrap:** Framework basado en JavaScript, del que hemos hecho uso de gran parte de sus componentes (barras de navegación, tablas, mensajes, paneles, etc.) para la elaboración de las vistas.
- **Thymeleaf:** Mediante la cual ha sido posible el desarrollo de las vistas basado en plantillas. Dada la versatilidad de este lenguaje ha sido mucho más cómodo y sencillo la implementación del front-end.

Para el back-end:

- **Spring:** Framework de Java para el desarrollo de aplicaciones web. Lo hemos utilizado tanto para la aplicación base, como para Quiz.
- **JPA:** API de Persistencia de Java, la cual implementa el patrón Almacén del Dominio. Ocupándose así de la persistencia y de la integración de la aplicación.
- **Maven:** Apache Maven es una herramienta que tiene por base tanto la auto-construcción del software como la gestión de dependencias del mismo.
- **MySQL:** Lenguaje de bases de datos relacionales, utilizado para el almacenamiento relativo a la capa de datos de nuestro proyecto.

Exposición de proyectos en entorno virtual:

- **VirtualBox:** Herramienta para la virtualización y ejecución de sistemas operativos. En nuestro caso Ubuntu, basado en Linux.

- **Vagrant:** Herramienta de construcción, configuración y mantenimiento de entornos virtuales.
- **Docker:** Esta herramienta nos permite ejecutar aplicaciones desde entornos virtuales.

Control de Versiones:

- **GitHub:** Herramienta de Control de Versiones basada en Git. La hemos utilizado en su versión web.
- **Sourcetree:** Aplicación de Control de Versiones basada en Git. Esta ha sido la aplicación de escritorio que hemos utilizado para el control de versiones, dado que nos ha resultado una herramienta muy cómoda, visual y sencilla para gestionar las diferentes versiones del proyecto.
- **Google Drive:** Herramienta web, que nos ha permitido almacenar los elementos de la configuración del software relativos a la documentación, de forma compartida. Google Drive versiona todos los elementos subidos, esto nos permite llevar, del mismo modo, un control de las versiones que se hayan dado.

Arquitectura Software:

- **Dia:** Herramienta para la generación de diagramas. Utilizada en nuestro caso para la definición de los Diagramas de Clases UML, como para los de Modelo del Dominio.
- **Visual Paradigm:** Utilizada en su versión on-line para la generación de los Diagramas de Componentes. Permite generar otros tipos de diagramas como UML, E-R, etc.

Gestión de tareas:

- **Trello:** Herramienta dedicada a la gestión y asignación de tareas, mediante un tablero dividido en diferentes estados.

Pruebas de peticiones API REST:

- **Postman:** Aplicación de testeo de las diferentes consultas de una API REST.

Definición de las peticiones API REST:

- **Swagger Editor:** Aplicación de definición de las peticiones API y generación de las mismas.

4.5. Fases del desarrollo

En este apartado se explicará y detallará cómo ha sido todo el proceso de desarrollo del software desde sus fases iniciales hasta su finalización.

4.5.1. Fase de planificación

Las primeras fases del proyecto han consistido en una primera toma de contacto, por parte del equipo de desarrollo, con la temática principal del Proyecto de Fin de Grado, en este caso, Plataforma de Retos, Gamificación y Analíticas.

Los primeros puntos que se trataron durante la planificación del proyecto fueron:

- En qué consistía la aplicación.
- ¿Qué tenía que hacer la aplicación?
- ¿Quiénes son los usuarios a los que va destinado el software final?

Todo ello queda detallado en el punto [2. Objetivos](#). Una vez definidos se estableció que la metodología a utilizar iba a ser combinada entre Scrum y Kanban, ya que se consideró que ambas tenían características interesantes que serían de gran utilidad durante el desarrollo de la plataforma, ver apartado [3.1. Metodología de desarrollo](#).

También, durante esta fase de planificación comenzamos con las comparativas entre las diferentes aplicaciones que serían integradas en nuestro proyecto, con el fin de poder escoger la mejor de ellas. En cuanto a las aplicaciones de retos se analizaron Quizziz, Kahoot!, Arsnova, estableciendo sus pros y sus contras, ver punto [1.1.2. Herramientas educativas basadas en retos](#). Como resultado de este análisis se estableció oportuno que la aplicación que más se ajustaba a las necesidades del proyecto era Arsnova.

De manera análoga se procedió a valorar los diferentes motores de gamificación, ver el apartado [1.2.1. Motores de gamificación](#), en el que se establece la diferencia entre Sugar Engine y ActiDoo. Finalmente se procedió a la elección de ActiDoo.

Durante la planificación, también se establecieron las herramientas y tecnologías con las que se iba a implementar la Plataforma RGA. Se decidió una implementación con el Framework de Java, Spring. Junto con HTML, CSS, MySQL, entre otros, ver punto [4.4. Tecnologías utilizadas](#). Se escogieron estos lenguajes de programación por ser conocidos por sendos integrantes del equipo de desarrollo.

Por último, pero no por ello menos importante, se estableció la arquitectura, detallada en el punto [4.2. Arquitectura del software de Plataforma RGA](#). Dado que se está hablando de un desarrollo en una metodología ágil, la arquitectura se estableció en un principio, pero se fue mejorando a medida que fue avanzando el proyecto.

4.5.2. Fase de prototipado

Una vez planificado el proyecto, se comenzó con la familiarización de los nuevos entornos de desarrollo, como por ejemplo Spring Tool Suite, para desarrollo de Spring Boot.

Para ello se llevó a cabo la implementación de diferentes prototipos funcionales, siendo estos:

- **Prototipo de interfaz:** Se implementó con el fin de conocer la estructura y funcionamiento básico de Spring. Así como el mapeo de las URLs establecidas en el controlador. En este caso, consistió en la visualización de una página de bienvenida.
- **Prototipo de formulario:** Se diseñó un prototipo, con un formulario con diversos campos a ser rellenados. De este prototipo se implementaron dos variantes, la primera no funcional, sin envío de los datos al servidor. Y la segunda, plenamente funcional con envío de los datos. La interfaz se muestra a continuación, en la figura 17.

The image shows a web form prototype on a dark gray background. At the top, the text "Prototipo Formulario" is displayed in a large, white, sans-serif font. Below this, there are four white rectangular input fields stacked vertically. The first field is labeled "Nombre y apellidos", the second "Ciudad", the third "País", and the fourth "Edad". All labels are in a small, gray, sans-serif font. At the bottom of the form, there is a wide, green rectangular button with the word "Enviar" in white, bold, sans-serif font.

Figura 17. Prototipo de formulario.

- **Prototipo de inicio de sesión:** Con el prototipo de formulario funcionando, se implementó el de inicio de sesión con un campo de usuario y otro de contraseña, con el fin de poder autenticarse. Ver figura 18.
- **Prototipo de registro:** De forma complementaria al anterior, se realizó un prototipo de registro para poder darse de alta en una plataforma. Como se muestra en la figura 19.



Prototipo
Iniciar Sesión

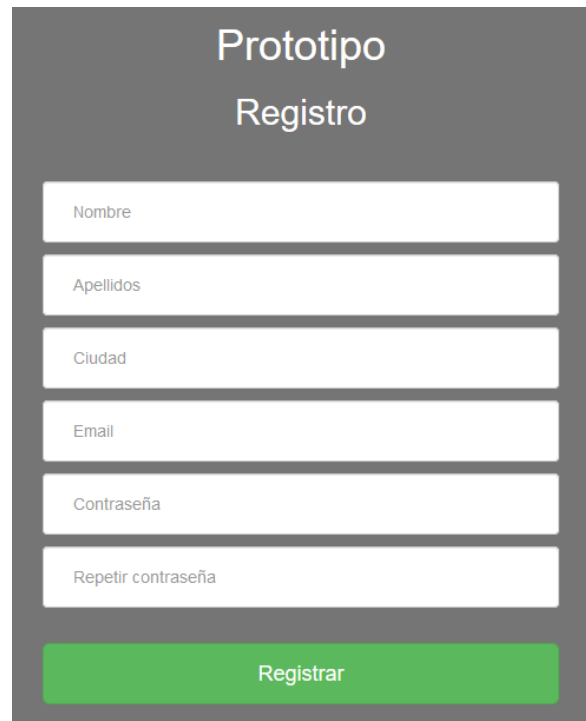
Usuario

Contraseña

Iniciar Sesión

☐ Recordar contraseña

Figura 18. Prototipo de inicio de sesión.



Prototipo
Registro

Nombre

Apellidos

Ciudad

Email

Contraseña

Repetir contraseña

Registrar

Figura 19. Prototipo de registro.

4.5.3. Fase de implementación

Con los prototipos ya desarrollados y la arquitectura bien definida, se comenzó con la implementación del proyecto. Para ello, lo primero que se definieron fueron las entidades especificadas en el modelo del dominio, ver apartado [4.2. Arquitectura del software de Plataforma RGA](#). A la par del desarrollo Java, se fue implementando toda la estructura JPA, encargada de representar el Patrón Almacén del Dominio. Asimismo, se ha hecho uso de JPA para reducir el tamaño de la capa de integración y simplificando la arquitectura y dimensiones del software.

De las cuantiosas anotaciones de JPA que se han utilizado en el proyecto, las tres más destacadas han sido:

- **@Entity**: Marca una clase como entidad del proyecto. La entidad representa además una tabla en la base de datos y cada uno de sus atributos es una de las columnas. Las clases de entidad anotadas han sido, Usuario, Alumno, Profesor, Asignatura y Reto.
- **@Service**: Esta anotación indica que la clase que estamos tratando es un Servicio de Aplicación. De modo que esta contendrá toda la lógica de negocio, además de los métodos CRUD, *create*, *read*, *update* y *delete*.
- **@Repository**: La cual establece que la clase anotada ejercerá de repositorio JPA, siendo el encargado de acceder a la base de datos.

Tras todo el desarrollo de la capa de negocio, la estructura simplificada resultante del proyecto de la Plataforma RGA sería la siguiente:

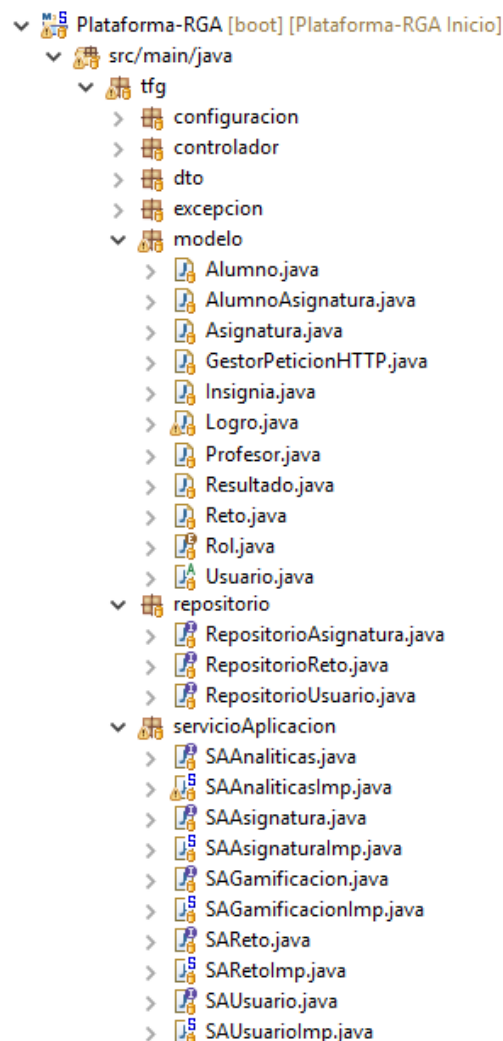


Figura 20. Esquema simplificado de la estructura de Plataforma RGA.

En cuanto a la implementación de las vistas, se reutilizaron los prototipos de inicio de sesión y registro mencionados previamente. Para los que se empleó Spring Security, haciendo así la aplicación web más segura.

Para el resto de vistas internas de la aplicación se hizo uso de Bootstrap, principalmente. Las interfaces de usuario resultantes de este desarrollo quedan detalladas en el [Anexo II: Manual de usuario](#).

4.5.4. Fase de integración

Una vez superados los inconvenientes de Arsnova y ActiDoo citados en el apartado [4.1. Evolución del desarrollo del proyecto y descripción de la solución final integrada](#), se integraron tanto Quiz, como Sugar Engine, respectivamente.

El desarrollo de la aplicación de retos, Quiz, con la API y a medida de los requerimientos de la Plataforma RGA, queda detallado en el siguiente apartado, [4.5.5. Implementación e integración de Quiz](#).

Con la herramienta de gamificación de Sugar Engine y su integración, se pudieron desarrollar elementos como los mencionados a continuación, entre otros:

- Creación de usuarios: Dar de alta los alumnos que participen en un reto.
- Creación de juegos: Crear un juego, del que aplicar la gamificación.
- Creación de insignias: Crear las insignias que podrán ganar los alumnos.
- Creación de rankings: Donde aparecerán las puntuaciones y las insignias de los alumnos.
- Actualizar puntuaciones: De los alumnos, a medida que vayan progresando.
- Recoger premios: De los alumnos que hayan cumplido determinados objetivos.
- Obtener rankings: Con el fin de poder visualizar la clasificación del alumnado.

Durante la integración de Sugar Engine hubo que realizar numerosas pruebas sobre las peticiones de la API, ya que en ocasiones la documentación oficial carecía de suficiente nivel de detalle.

Un ejemplo es la autenticación inicial contra la API, de la cual es necesario recoger un Bearer Token con el código de la autenticación. Una vez obtenido, era necesario añadirlo a la cabecera de cada petición para que esta fuese satisfactoria. Nada de esto fue encontrado en la documentación, y sólo pudo ser hallado y averiguado haciendo pruebas sobre la pestaña de Red del Navegador, conjuntamente con la aplicación Postman.

Por último, cabe destacar que a medida que iba avanzando el desarrollo se fueron probando y testeando los diferentes métodos y funcionalidades de forma manual. Del mismo modo que las pruebas de aceptación, de sistema, de interfaz, etc. se llevaron a cabo paulatinamente durante la implementación del proyecto.

4.5.5. Implementación e integración de Quiz

Para el desarrollo de Quiz se llevó a cabo la misma metodología, sistema de planificación, estructura, herramientas, tecnologías, etc. Con el fin de mantener la complejidad técnica y no añadir grandes cambios al proyecto.

Para su desarrollo, se utilizó Spring Boot, al igual que con la Plataforma RGA, siguiendo el esquema de arquitectura que definimos para Quiz, tras la decisión de tener que implementar una aplicación propia, a medida. Siendo su estructura de clases y paquetes Java la siguiente:

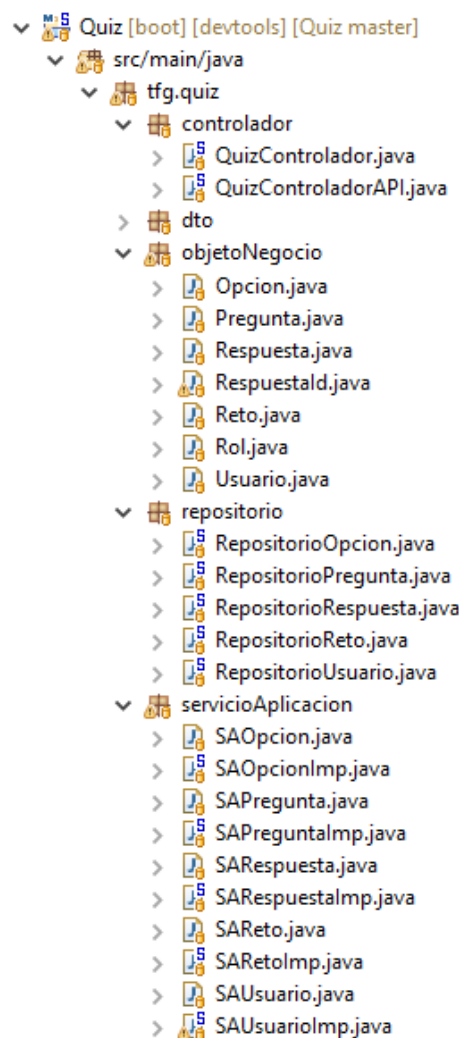


Figura 21. Esquema de la estructura de Quiz.

Además, se implementó una API REST a la que poder referenciar desde la Plataforma RGA, con peticiones como las que se muestran a continuación:

- Crear reto: De tal manera que se cree un reto en la aplicación.
- Obtener resultados: Con el fin de obtener todos los resultados pertenecientes a un mismo reto.
- Eliminar reto: Borra el reto del identificador que se incluya en la petición.

Toda la documentación de las peticiones de la API queda reflejada en el [Anexo IV: Referencia de la API de Quiz](#). Del mismo modo se tuvo que implementar una API REST en la Plataforma RGA, para ser usada desde Quiz, cuyas características figuran en el [Anexo III: Referencia de la API de Plataforma RGA](#).

4.5.6. Implementación del módulo de analíticas

El último módulo añadido a la Plataforma RGA fue el de analíticas. Para ello se implementó un registro que salvase todas y cada una de las respuestas del alumnado en cada reto. Almacenando así información relevante como el nombre del reto, la pregunta, la opción marcada, un booleano indicando si es correcta o no, los diferentes intervalos de tiempo de respuesta, etc. Esto se llevó a cabo mandando una petición a la API de la aplicación Quiz, devolviendo los resultados de toda una clase, y mostrándolos por pantalla en la Plataforma RGA. Tal y como se muestra en la siguiente figura:

Alumno	Reto	Pregunta	Respuesta	Correcta	Tiempo
Luis Rodríguez	Prueba Inicial	Indica cuál no es un lenguaje de BD Relacionales	MongoDB	✓	8.0 s
Luis Rodríguez	Prueba Inicial	Para seleccionar todas las filas devueltas de una consulta utilizaremos:	SELECT	✗	11.0 s
Luis Rodríguez	Prueba Inicial	¿Qué consulta usaremos para borrar la tabla Empleados?	DELETE TABLE Empleados	✗	6.0 s
Luis Rodríguez	Prueba Inicial	¿Cuál es el conector de BD de Java?	JDBC	✓	9.0 s
Luis Rodríguez	Prueba Inicial	Para actualizar utilizaremos:	UPDATE	✓	6.0 s
Carolina Sánchez	Prueba Inicial	Indica cuál no es un lenguaje de BD Relacionales	MongoDB	✓	9.0 s
Carolina Sánchez	Prueba Inicial	Para seleccionar todas las filas devueltas de una consulta utilizaremos:	SELECT *	✓	12.0 s
Carolina Sánchez	Prueba Inicial	¿Qué consulta usaremos para borrar la tabla Empleados?	DELETE TABLE Empleados	✗	7.0 s
Carolina Sánchez	Prueba Inicial	¿Cuál es el conector de BD de Java?	CBDJ	✗	8.0 s
Carolina Sánchez	Prueba Inicial	Para actualizar utilizaremos:	UPDATE	✓	14.0 s
Lorena Sanz	Prueba Inicial	Indica cuál no es un lenguaje de BD Relacionales	Oracle SQL	✗	18.0 s

Figura 22. Tabla de contenido de resultados analíticos.

Además del registro de respuestas anterior se desarrollaron una serie de gráficos analíticos con el fin de poder visualizar diferentes tendencias y patrones en la respuesta de los alumnos. El primero de ellos es un gráfico de líneas en el que se puede ver el tiempo de respuesta de los alumnos de la clase y su cómputo de tiempo global con respecto al reto.

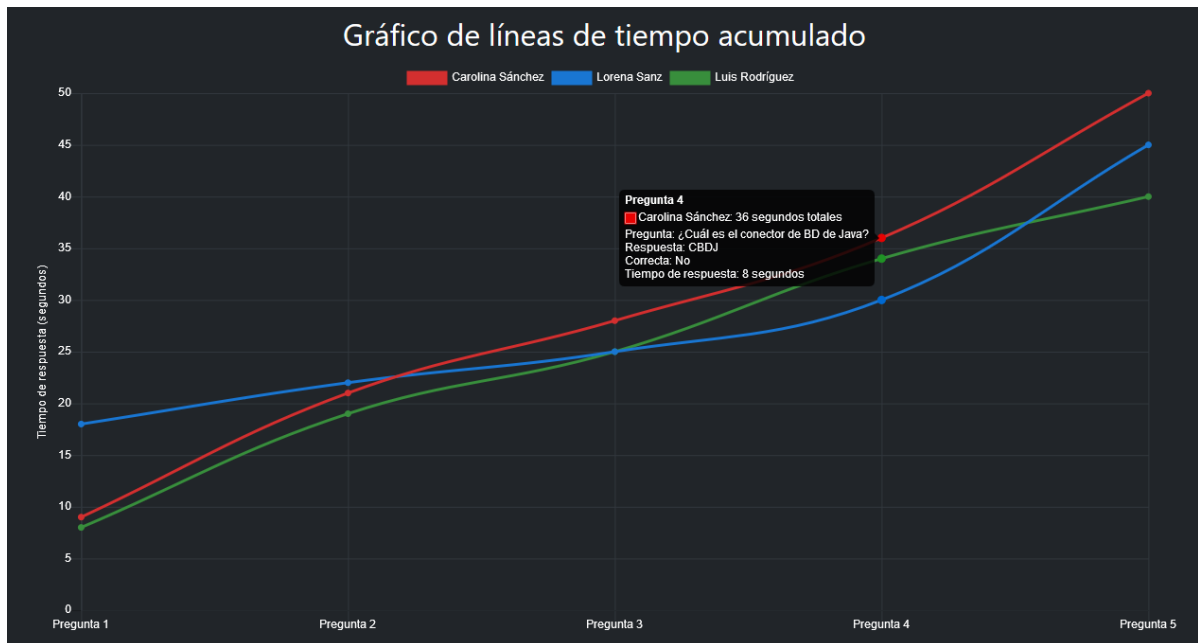


Figura 23. Gráfico de líneas de tiempo acumulado.

Así mismo, al tratarse de una aplicación enfocada al apoyo académico del alumnado, el profesor podrá filtrar los datos de manera individualizada para cada alumno. Ver la figura 24.



Figura 24. Gráfico de líneas de tiempo acumulado, filtrado.

También se han implementado gráficos de barras que muestran el tiempo de respuesta de cada pregunta. Pudiendo así obtener trazas o tendencias, además de en el mencionado tiempo, del nivel de certitud que tienen los alumnos sobre una misma cuestión. Por ejemplo, en la siguiente figura se puede observar que la pregunta número tres les ha costado menos responderla y por tanto estaban más seguros de su respuesta que en la pregunta cinco.



Figura 25. Gráfico de barras de tiempo de respuesta.

Por último, pero no por ello menos importante el presente módulo de analíticas dispone de un gráfico de barras apiladas que podrá ser de gran ayuda al profesorado, ya que muestra el porcentaje de aciertos para una pregunta. De modo que se pueda conocer cuáles son las cuestiones o temas en los que más fallan los estudiantes.

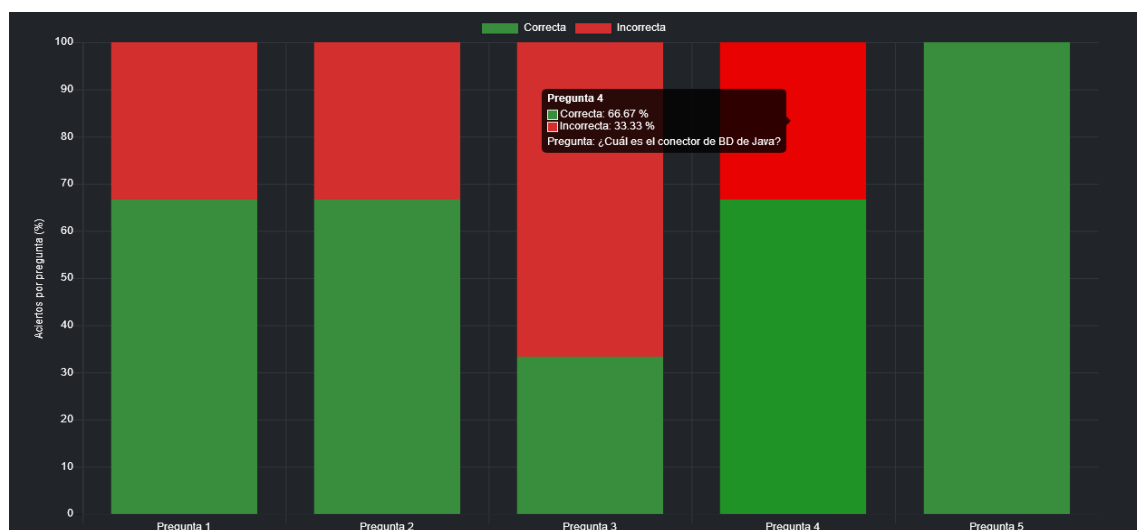


Figura 26. Gráfico de barras de aciertos.

Además podrá ser filtrado, como en todos los gráficos añadidos. En este caso, filtrando por respuestas incorrectas se generará el gráfico que se muestra en la figura 27.



Figura 27. Gráfico de barras de aciertos, filtrado.

Cabe mencionar que los gráficos de este módulo han sido implementados con la librería Chart.js, de JavaScript, encargada de generar gráficos mediante una entrada formateada de datos. Su sitio web es <https://www.chartjs.org/>.

4.6. Resultados finales

Como resultado final, tras el desarrollo del proyecto se ha obtenido un producto software con una aplicación base, Plataforma RGA, que es capaz de aunar e integrar la plataforma de retos Quiz, en la que se resuelven las cuestiones propuestas por el profesor; el motor de gamificación procesando y gamificando los resultados de los retos; y por último una base analítica sobre la que poder trazar estadísticas a partir de los resultados que esta arroja. Todo ello sumado a la propia gestión de usuarios, asignaturas y retos de la propia aplicación base.

Por último, cabe destacar la seguridad de la que gozan tanto las aplicaciones de Quiz como la Plataforma RGA, dado a que ambas han sido implementadas bajo Spring Security.

5. Aportaciones personales

En este apartado se resumirán y detallarán las aportaciones personales de los diferentes integrantes de este proyecto.

5.1. Carlos Marrón Díaz

En las fases iniciales del proyecto establecimos cuáles eran las herramientas y tecnologías que íbamos a utilizar.

Dado que nuestro proyecto se iba a tratar de una aplicación web, y además establecimos que la programaríamos en un lenguaje que conociésemos ampliamente ambos integrantes del equipo de desarrollo. El entorno de desarrollo (IDE) que establecimos fue Spring Tool Suite, para desarrollar aplicaciones Spring Boot, en el lenguaje de programación Java. A pesar de que el lenguaje de programación sí lo conocíamos, las primeras semanas de desarrollo nos centramos en investigar y aprender el uso del mencionado IDE y las características principales de Spring Boot.

Posteriormente continuamos con el desarrollo de determinadas maquetas y prototipos funcionales, con el fin de dar uso a los conocimientos adquiridos las semanas anteriores. Estos fueron:

1. Creación de vistas sencillas.
2. Formularios sencillos sin peticiones al servidor.
3. Formularios con peticiones al servidor y almacenamiento en la base de datos.
4. Creación de una página de inicio de sesión.
5. Creación de página de registro.

Una vez sentados estos conocimientos principales pasamos a establecer la planificación, y metodología de desarrollo. Para esta última descartamos tradicionales, aplicando una metodología ágil detallada en el punto [3.1. Metodología de desarrollo](#). Así mismo la planificación se llevó a cabo en otra de las herramientas que ya conocíamos, Trello, descrita en el punto [3.2. Planificación del proyecto](#).

Cuando ya estaban las bases del proyecto colaboré en el desarrollo de la Plataforma RGA.

La arquitectura también se definió desde el principio del proyecto y se fue mejorando a medida que fue avanzando el desarrollo. Ver punto [4.2. Arquitectura del software de Plataforma RGA](#).

Durante el desarrollo de la aplicación base se insertaron:

- Gestión de asignaturas.
- Gestión de retos.
- Gestión de alumnos.
- Aplicación preparada para la integración de una plataforma de retos.
- Aplicación preparada para la integración de un motor de gamificación.

Una vez finalizada esta primera parte de la Plataforma RGA, nos repartimos las tareas, mi parte consistía en integrar la aplicación de retos Arsnova a la nuestra, ya creada previamente. Después de que realizaré diversas pruebas para comprobar cómo funcionaba y de bastantes intentos para poder acoplarlo a nuestro proyecto fue imposible, ya que había que profundizar bastante en el código y adaptarlo a nuestras necesidades, ya que no disponía de API.

Una vez presentado el problema de Arsnova y comprobar que el motor de gamificación elegido era demasiado simple, decidimos cambiar el motor de gamificación Actidoo por el de Sugar Engine. En cuanto a la aplicación de retos, decidimos crear una desde cero. Una vez llegado a este punto decidimos cambiar los papeles, por lo que en este caso me tocó acoplar el motor de gamificación a nuestro proyecto.

Este motor mostraba ciertas diferencias respecto al anterior, ya que para que pudiera funcionar había que mandar una petición iniciando sesión como administrador y de su respuesta hay que coger la autorización e introducirla en todas las cabeceras.

Para comprobar el estado de las peticiones, así como sus parámetros y cabeceras utilicé Postman. De estas peticiones a la API REST de Sugar Engine, se hicieron uso de aquellas que aportaban la funcionalidad que requería nuestro proyecto, las cuáles son:

- Crear usuario.
- Crear, inicializar y actualizar el leaderboard.
- Crear reto.
- Crear y mostrar el logro o insignia.
- Dar el premio.

Por último, cabe destacar que también participé en llevar a cabo las diferentes pruebas manuales del proyecto. Las reuniones periódicas con Jorge y con el director del proyecto, así como en la elaboración del presente documento.

5.2. Jorge Merino Martínez

En las fases iniciales del proyecto establecimos cuáles eran las herramientas y tecnologías que íbamos a utilizar.

Dado que nuestro proyecto se iba a tratar de una aplicación web, y además establecimos que la programaríamos en un lenguaje que conociésemos ampliamente ambos integrantes del equipo de desarrollo. El entorno de desarrollo (IDE) que establecimos fue Spring Tool Suite, para desarrollar aplicaciones Spring Boot, en el lenguaje de programación Java. A pesar de que el lenguaje de programación sí lo conocíamos, las primeras semanas de desarrollo nos centramos en investigar y aprender el uso del mencionado IDE y las características principales de Spring Boot.

Posteriormente continuamos con el desarrollo de determinadas maquetas y prototipos funcionales, con el fin de dar uso a los conocimientos adquiridos las semanas anteriores. Estos fueron:

1. Creación de vistas sencillas.
2. Formularios sencillos sin peticiones al servidor.
3. Formularios con peticiones al servidor y almacenamiento en la base de datos.
4. Creación de una página de inicio de sesión.
5. Creación de página de registro.

Una vez sentados estos conocimientos principales pasamos a establecer la planificación, y metodología de desarrollo. Para esta última descartamos tradicionales, aplicando una metodología ágil detallada en el punto [3.1. Metodología de desarrollo](#). Así mismo la planificación se llevó a cabo en otra de las herramientas que ya conocíamos, Trello, descrita en el punto [3.2. Planificación del proyecto](#).

Ya con las bases del proyecto marcadas me dediqué al desarrollo de la Plataforma RGA, incorporando tanto la página de inicio de sesión como la página de registro.

La arquitectura también se definió desde el principio del proyecto y se fue mejorando a medida que fue avanzando el desarrollo. Ver punto [4.2. Arquitectura del software de Plataforma RGA](#).

Durante el desarrollo de la aplicación base se insertaron:

- Gestión de asignaturas.
- Gestión de retos.
- Gestión de alumnos.
- Aplicación preparada para la integración de una plataforma de retos.
- Aplicación preparada para la integración de un motor de gamificación.

Una vez finalizada esta primera parte de la Plataforma RGA, inicié la integración del motor de gamificación de ActiDoo, con características como el alta de un usuario, incremento del progreso, obtención de las insignias. Pero dados los problemas que surgieron [1.2.1. Motores de gamificación](#), este no nos sirvió, teniendo así que desecharlo, aunque mantuvimos la estructura de la integración. En concreto el mapeo de las URLs del Controlador y la estructura básica del Servicio de Aplicación de Gamificación, tanto la interfaz como la implementación.

Por otro lado también me encargué del desarrollo y la implementación de Quiz, dado que integrar Arsnova no fue posible. En el desarrollo de Quiz, seguimos con los mismos paradigmas y principios que en el resto del proyecto, metodología ágil, planificación, etc.

Se desarrollo con las siguientes funcionalidades:

- Alta de usuarios, verificados por token, ya sean alumnos o profesores.
- Alta y modificación de retos con sus respectivas preguntas.
- Desarrollo de la API.
- Integración con Plataforma RGA.

Finalmente, y pese a que en un principio pensábamos que no nos iba a dar tiempo a llevarlo a cabo por falta de tiempo, llevamos a cabo la parte de analíticas. La cual consiste en una tabla que recopila todos los datos relativos a las respuestas de los usuarios, ya sea:

- El reto al que pertenece la respuesta.
- La pregunta en cuestión.
- La opción marcada.
- La opción correcta.
- El intervalo de tiempo de respuesta.
- Un booleano que indica la veracidad de la respuesta.

Del módulo de analíticas también me encargué de la integración de la librería Chart.js y de la generación de los gráficos de barras, líneas, barras apiladas, etc. Con el fin de llevar a cabo un seguimiento mayor del alumno y de su progreso. Además, esta es una de las partes que quedaría para [8. Trabajo futuro](#) pudiéndose extender a campos más propios de la estadística o la matemática, con generación de funciones, progresos, medias, desviaciones típicas, etc.

Por último, cabe destacar que también participé en llevar a cabo las diferentes pruebas manuales del proyecto. Las reuniones periódicas con Carlos y con el director del proyecto, así como en la elaboración del presente documento.

6. Conclusiones

Como resultado de este trabajo de fin de grado, se ha desarrollado una plataforma que integra los Retos, Gamificación y Analíticas, dado que ninguna de las analizadas previamente, Kahoot!, Quizziz, Arsnova, etc. conseguían aglutinar estas tres funcionalidades y menos proponiendo una solución modular que permitiera cambiar alguno de los módulos.

Como resultado se ha conseguido desarrollar una aplicación funcional que integre una plataforma de retos, en este caso implementada por nosotros, Quiz; un motor de gamificación, Sugar Engine; y a la vez un registro analítico con el fin de extraer resultados del progreso del alumnado.

Asimismo, se ha desarrollado, tal y como se indica en el texto anterior, la aplicación Quiz, la cual además de poseer una interfaz de usuario, también dispone de una API REST. Pudiendo hacer uso de esta aplicación de forma remota, a través de peticiones.

Por todo lo expuesto anteriormente, esta Plataforma de Retos, Gamificación y Analíticas puede resultar una herramienta muy interesante para su uso en la docencia o en otros ámbitos, como laborales, marketing, etc. con el fin de formar de una manera dinámica, entretenida y diferente a los participantes, sin necesidad de que la enseñanza sea vista como algo aburrido y puramente teórico.

En lo que al equipo de desarrollo se refiere, este proyecto nos ha sido de gran utilidad para poder conocer nuevas herramientas como Spring Boot, principalmente. Así como la puesta en práctica de conocimientos estudiados a lo largo del Grado en Ingeniería del Software, como puedan ser los diagramas de clases UML y de modelo del dominio, la aplicación y uso de herramientas como JPA, o el uso de Git como sistema de control de versiones.

7. Conclusions

The main result of this work is an open platform that integrate three main proposed elements Challenges, Gamification and Analytics, due to none of the previously analyzed such as Kahoot!, Quizziz, Arsnova, etc. didn't provide an integrated solution, let alone, a modular one that allows to interchange some of the modules.

As a result, it has been possible to develop a functional application that integrates a challenges platform, in this case implemented by us: Quiz; a gamification engine, Sugar Engine; and at the same time an analytical register that extracts results of the students' progress.

Also, as indicated in the previous paragraph the Quiz application provides a REST API that makes it able to be used remotely throughout requests.

AS a result, this Challenges, Gamification and Analytics platform can be a very interesting tool to be use by teacher as a supporting tool in class, but also can be applied in other areas, such as work, marketing, etc. to form in a dynamic, entertaining and different way the participants making the teaching be seen as boring and purely theoretical.

As far as the development team is concerned, this project has been very useful for us to know new tools such as Spring Boot, mainly. In this we have put into practice the knowledge of the programs studied throughout the Software Engineering Degree, such as the UML programs and the domain model, the application and use of tools such as JPA, or the use of Git as version control system.

8. Trabajo futuro

Tras el desarrollo software de la Plataforma de Retos, Gamificación y Analíticas, existen algunos elementos y opciones que serían interesantes poder implementar como trabajo futuro, ampliando así la funcionalidad de la aplicación.

Una de ellas sería la opción de poder repetir los retos, sin necesidad de que los lance de nuevo el profesor. De modo que pudiese valer a los alumnos para poder practicar y estudiar para los exámenes. Y con ello ampliar las analíticas para poder conocer el interés de los alumnos en la materia.

En la parte de retos también se podrían añadir desafíos entre dos alumnos y que se reten, pudiendo premiar al ganador con puntos o insignias. De la misma manera también sería interesante la implementación de torneos para toda una misma asignatura.

Además, actualmente los retos en Quiz están limitados a preguntas de respuesta simple, por lo que sería interesante ampliar la herramienta para permitir otros tipos de preguntas como preguntas de respuesta múltiple, de secuencia, etc. Por otro lado, se podrían flexibilizar las puntuaciones de los retos, pudiendo dar la opción de elegir si se quiere puntuar por la rapidez de respuesta.

Por último, se podría añadir que, al realizar el reto, se muestre la opción de realizarlo por parejas o grupos. La idea sería que un único miembro del equipo interactuara con la herramienta, pero aplicando la puntuación entre todos los participantes.

9. Future work

Through the development of the Challenges, Gamification and Analytics Platform, there are some elements and functionalities that would be interesting to implement for a future work, and in this way expand the application functionality.

One of them would be the option of being able to repeat the challenges, without the teacher launching them again. This way the students could use it to practice and study for the exams. In addition, the ability to repeat challenges would also allow to expand the gathered analytic data in order to know the interest of the students about the subject.

Regarding new features for challenges, it would be interesting to allow students to challenge each other and the winner could be rewarded with points or badges. It would also be interesting the implementation of tournaments for the same subject.

In addition, currently quizzes are limited to questions with simple answers (just one), it would be nice to allow teachers to create quizzes with different type of questions, for example, multiple choice, sequence, etc. Besides, would be nice to add some flexibility to the score calculation, based on the time used to answer the questions.

Finally, it would be interesting to allow the option to create challenges for pairs or groups of students. The idea would be to allow one student to represent the entire group of students within the tool, but applying the score to the entire group of students.

Bibliografía

- [1]. Ciampa, K. (2014), Learning in a mobile age. *Journal of Computer Assisted Learning*, 30: 82-96. doi:10.1111/jcal.12036
- [2]. KLEIN, H. J., NOE, R. A. and WANG, C. (2006), MOTIVATION TO LEARN AND COURSE OUTCOMES: THE IMPACT OF DELIVERY MODE, LEARNING GOAL ORIENTATION, AND PERCEIVED BARRIERS AND ENABLERS. *Personnel Psychology*, 59: 665-702. doi:10.1111/j.1744-6570.2006.00050.x
- [3]. Malone, T. W., & Lepper, M. R. (1987). Making learning fun: A taxonomy of intrinsic motivations for learning. *Aptitude, learning, and instruction*, 3(1987), 223-253
- [4]. Marina Papastergiou, Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation, *Computers & Education*, Volume 52, Issue 1, 2009, Pages 1-12, ISSN 0360-1315, <https://doi.org/10.1016/j.compedu.2008.06.004>.
- [5]. Juan C. Burguillo, Using game theory and Competition-based Learning to stimulate student motivation and performance, *Computers & Education*, Volume 55, Issue 2, 2010, Pages 566-575, ISSN 0360-1315, <https://doi.org/10.1016/j.compedu.2010.02.018>.
- [6]. Fies, C. & Marshall, J. J *Sci Educ Technol* (2006) 15: 101. <https://doi.org/10.1007/s10956-006-0360-1>
- [7]. Johns, K. (2015). Engaging and assessing students with technology: A review of kahoot! *Delta Kappa Gamma Bulletin*, 81(4), 89-91. Retrieved from <http://0-search.proquest.com/cisne.sim.ucm.es/docview/1706873637?accountid=14514>
- [8]. Y. Chaiyo and R. Nokham, "The effect of Kahoot, Quizizz and Google Forms on the student's perception in the classrooms response system," 2017 International Conference on Digital Arts, Media and Technology (ICDAMT), Chiang Mai, 2017, pp. 178-182.
- [9]. Ferguson, Rebecca. *The State of Learning Analytics in 2012: A Review and Future Challenges*. Technical Report. Knowledge Media Institute: The Open University, UK, 2012.

- [10].G. Siemens, G., Dawson, S., & Lynch, Improving the Quality and Productivity of the Higher Education Sector. Policy and Strategy for Systems-Level Deployment of Learning Analytics., (2013)
- [11].Top 3 alternativas a Kahoot (2017, 29 de agosto), [en línea]. Disponible en: <http://el-proyector-de-clase.blogspot.com/2017/08/3-alternativas-kahoot.html> [2018, 05 de Agosto].
- [12].Christian Delgado von Eitzen (2017). Plickers: qué es y cómo usar esta genial herramienta gratis en clase, [en línea]. Disponible en: <http://www.christiandve.com/2017/07/plickers-que-es-usar-clase-herramienta/> [2018, 05 de Agosto].
- [13].González, C. (2014, 13 de mayo). Estrategias de gamificación aplicadas a la educación y la salud [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=f-q3jz5ukVI>
- [14].Cortizo Pérez, J.C., Carrero García, F., Monsalve Piqueras, B., Velasco Collado, A., Díaz del Dedo, L.I., Pérez Martín, J. (2011). Gamificación y Docencia: Lo que la Universidad tiene que aprender de los Videojuegos, [en línea]. Disponible en: [http://abacus.universidadeuropea.es/bitstream/handle/11268/1750/46 Gamificacio.n.pdf?sequence=2&isAllowed=y](http://abacus.universidadeuropea.es/bitstream/handle/11268/1750/46_Gamificacio.n.pdf?sequence=2&isAllowed=y) [2018, 21 de Agosto].
- [15].Carlos Mora (2015). Qué es, mecánicas y dinámicas de la gamificación, [en línea]. Disponible en: <https://gamificationedufis.wordpress.com/2015/05/27/que-es-mecanicas-y-dinamicas-de-la-gamificacion/> [2018, 21 de Agosto].
- [16].Santiago Moll (2014). Gamificación: 7 claves para entender qué es y cómo funciona, [en línea]. Disponible en: <https://justificaturespuesta.com/gamificacion-7-claves-para-entender-que-es-y-como-funciona/> [2018, 08 de Agosto].
- [17].Pablo Colado (2017). ¿Cuál es el país con más smartphones del mundo?, [en línea]. Muy interesante. Disponible en: <https://www.muyinteresante.es/tecnologia/preguntas-respuestas/cual-es-el-pais-con-mas-smartphones-del-mundo-101478776774> [2018, 22 de Agosto].
- [18].Liliana Arroyo (2016). ¿Qué son las analíticas de aprendizaje y por qué pueden revolucionar el mundo educativo?, [en línea]. Disponible en: <http://eldiariodelaeducacion.com/blog/2016/11/07/las-analiticas-aprendizaje-pueden-revolucionar-mundo-educativo/> [2018, 16 de Agosto].

- [19].Serrano, A. (2017). Mejorando la evaluación de juegos serios mediante el uso de analíticas de aprendizaje (tesis doctoral). Universidad Complutense, Madrid, España.
- [20].Williams, S., & Williams, N. (2003). The Business Value of Business Intelligence. *Intelligence*, 8 (301), 30–39.
- [21].Liñán, L. C., & Pérez, Á. A. J. (2015). Educational data mining and learning analytics: differences, similarities, and time evolution. *Rusc*, 12 (3), 98–112.
- [22].Fausett, L. V., & Elwasif, W. (1994). Predicting performance from test scores using backpropagation and counterpropagation. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)* (Vol. 5, pp. 3398–3402). IEEE.
- [23].Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). *Intelligent tutoring systems*. Science (New York, N.Y.), 228 (4698), 456–462.
- [24].Nwana, H. (1990). Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, 4 (4), 251–277.
- [25].Chang, K., Beck, J., Mostow, J., & Corbett, A. (2006). A Bayes Net Toolkit for Student Modeling in Intelligent Tutoring Systems (pp. 104–113). Springer Berlin Heidelberg.
- [26].Yudelson, M. V, Medvedeva, O., Legowski, E., Castine, M., Jukic, D., & Crowley, R. S. (2005). Mining student learning data to develop high level pedagogic strategy in a medical ITS. *Proceedings of AAAI Workshop on Educational Data Mining*.
- [27].Hämäläinen, W., & Vinni, M. (2006). Comparison of Machine Learning Methods for Intelligent Tutoring Systems (pp. 525–534). Springer Berlin Heidelberg.
- [28].Hassler, M. (2010). *Web Analytics*. Redline Heidelberg, 3 (1), 1–14.
- [29].Burr, L., & Spennemann, D. H. R. (2004). Patterns of user behavior in university online forums. *International Journal of Instructional Technology and Distance Learning*, 1 (10), 11–18.
- [30].Romero, C., López, M. I., Luna, J. M., & Ventura, S. (2013). Predicting students' final performance from participation in on-line discussion forums. *Computers and Education*, 68, 458–472.
- [31].Dringus, L. P., & Ellis, T. (2005). Using data mining as a strategy for assessing asynchronous discussion forums. *Computers & Education*, 45 (1), 141–160.

- [32].Kim, J., Chern, G.,Feng, D., Shaw, E., & Hovy, E. (2006). Mining and Assessing Discussions on the Web through Speech Act Analysis. Proceedings of the ISWC06 Workshop on Web Content Mining with Human Language Technologies, (April 2016).
- [33].Hernández, J. A., Ochoa, A., Muñoz, J., & Burlaka, G. (2006). Detecting cheats in online student assessments using Data Mining. In Conference on Data Mining | DMIN (p. Vol. 6, 205).
- [34].Elias, T. (2011) *Learning Analytics: Definitions, Processes, Potential*. Disponible en: http://edutechwiki.unige.ch/en/Learning_analytics#Tracking_in_e-learning_platforms [2018, 20 de agosto].

Referencias

<https://www.vagrantup.com/intro/index.html>

<https://www.docker.com/what-docker>

<https://swagger.io/tools/swagger-editor/>

<https://jquery.com/>

<http://api.sugarengine.org/v1/>

<https://github.com/ActiDoo/gamification-engine>

<https://www.bunchball.com/gamification>

<https://www.educativa.com/blog-articulos/gamificacion-el-aprendizaje-divertido/>

<http://www.christiandve.com/2017/07/plickers-que-es-usar-clase-herramienta/>

<http://eldiariodelaeducacion.com/blog/2016/11/07/las-analiticas-aprendizaje-pueden-revolucionar-mundo-educativo/>

<https://www.youtube.com/watch?v=f-q3jz5ukVI>

http://abacus.universidadeuropea.es/bitstream/handle/11268/1750/46_Gamificacion.pdf?sequence=2&isAllowed=y

<https://justificaturespuesta.com/gamificacion-7-claves-para-entender-que-es-y-como-funciona/>

<https://www.muyinteresante.es/tecnologia/preguntas-respuestas/cual-es-el-pais-con-mas-smartphones-del-mundo-101478776774>

<https://gamificationedufis.wordpress.com/2015/05/27/que-es-mecanicas-y-dinamicas-de-la-gamificacion/>

<http://www.baeldung.com/registration-with-spring-mvc-and-spring-security>

<https://www.thymeleaf.org/doc/tutorials/2.1/usingthymeleaf.html>

<https://material.io/>

<https://getbootstrap.com/docs/4.0/getting-started/introduction/>

<https://docs.spring.io/spring-security/site/docs/5.0.0.RELEASE/reference/htmlsingle/>

<https://plugins.jquery.com/cookie/>

<https://docs.spring.io/spring-data/jpa/docs/1.5.0.RELEASE/reference/html/jpa.repositories.html>

<https://github.com/google/gson>

<http://www.baeldung.com/spring-requestmapping>

<http://api.jquery.com/>

<https://www.gitignore.io/>

<https://viralpatel.net/blogs/spring-mvc-cookie-example/>

<https://vladmihalcea.com/the-best-way-to-map-a-many-to-many-association-with-extra-columns-when-using-jpa-and-hibernate/>

<https://hellokoding.com/jpa-many-to-many-extra-columns-relationship-mapping-example-with-spring-boot-maven-and-mysql/>

<https://spring.io/guides/tutorials/bookmarks/>

<http://www.objectdb.com/java/jpa/persistence/lock>

<http://www.chartjs.org/docs/latest/>

<https://www.visual-paradigm.com/tutorials/>

Anexo I: Manual de instalación

En este punto se va a explicar qué aplicaciones y herramientas se deben de instalar y los pasos que hay que llevar a cabo para poder ejecutar el proyecto realizado.

Para lanzar la aplicación, Spring Tool Suite (STS) sería el entorno de desarrollo necesario. Para poder descargarlo lo único que haría falta sería seguir el siguiente enlace <https://spring.io/tools> y descargar la versión correspondiente a nuestro sistema operativo de STS.

Para seguir con la instalación necesitamos instalar una máquina virtual, en nuestro caso hemos escogido Virtual Box la cual se puede descargar desde el siguiente enlace, <https://www.virtualbox.org/wiki/Downloads>. A parte de descargar VirtualBox también se recomienda la instalación de Oracle VirtualBox Extension Pack. Junto con VirtualBox será necesario descargar Vagrant e instalarlo.

Para poder continuar con la instalación y funcionamiento de la aplicación los siguientes puertos deben de estar liberados, para ser usados por sus respectivas aplicaciones:

- **3306:** Base de datos
- **8000:** Quiz
- **8081:** API de SugarEngine
- **8082:** Base de datos de SugarEngine
- **8083:** Panel de Administrador de SugarEngine
- **9000:** Plataforma RGA (Aplicación principal)

Los pasos necesarios para arrancar el proyecto serían los siguientes:

Para la base de datos:

- Poner en funcionamiento MySQL.
- Crear en el puerto 3306, relativo a la base de datos, una con nombre “tfg” y otra llamada “quiz”, ambas sin las comillas.

Para las aplicaciones Plataforma RGA y Quiz, bien podemos descargar los proyectos directamente desde los repositorios, e importarlos:

- <https://github.com/JorgeMerino/Plataforma-RGA>
- <https://github.com/JorgeMerino/Quiz>

O bien, importarlos a través de Spring Boot:

- Abrir el entorno de desarrollo STS, que se encuentra dentro de la carpeta descargada y ya descomprimida de Spring Tool Suite.
- Una vez ya abierto el programa, en la barra de herramientas seleccionar *Window/Show View/Other...*
- Seleccionar la carpeta *Git/Git repositories*
- Seleccionar la opción de *Clone a Repository/Github* y copiar la dirección de GitHub del repositorio, [JorgeMerino/Plataforma-RGA](#) y pulsar Search, escoger el proyecto en cuestión.

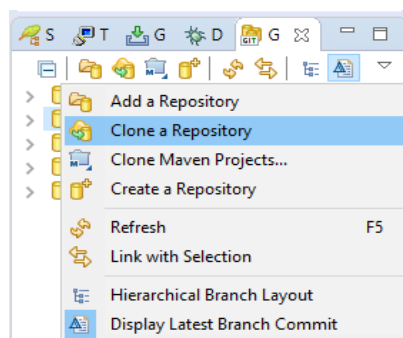


Figura 28. Clonar un repositorio.

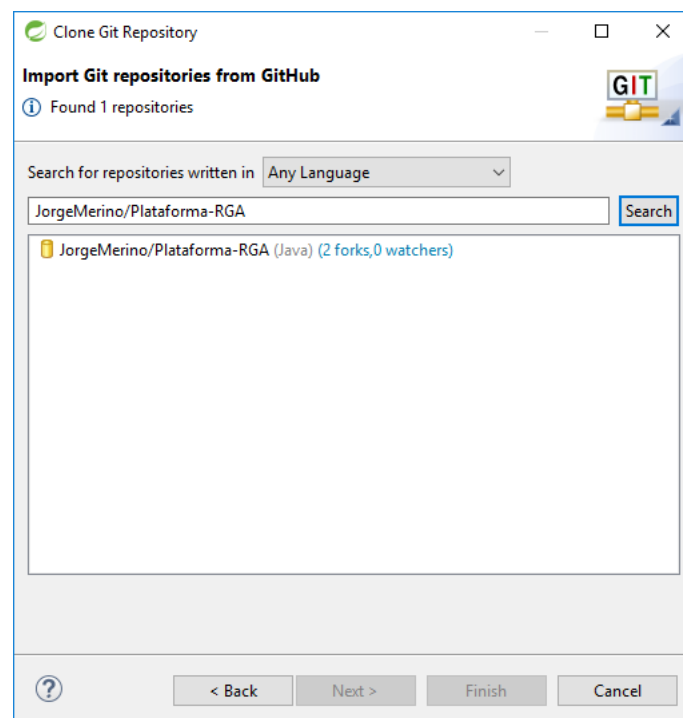


Figura 29. Seleccionar el repositorio.

- Importar desde la vista Git Repositories, haciendo clic derecho en el repositorio y pinchando en *Import Projects...*
- Una vez ya aparezca en el explorador de proyectos, lo seleccionamos y ejecutamos pinchando en *Run As/Spring Boot App*, tal y como se muestra en la figura 30.

Para importar el proyecto Quiz mediante Spring Boot se llevarán a cabo los mismos pasos mencionados anteriormente, cambiando la dirección del repositorio por [JorgeMerino/Quiz](#).

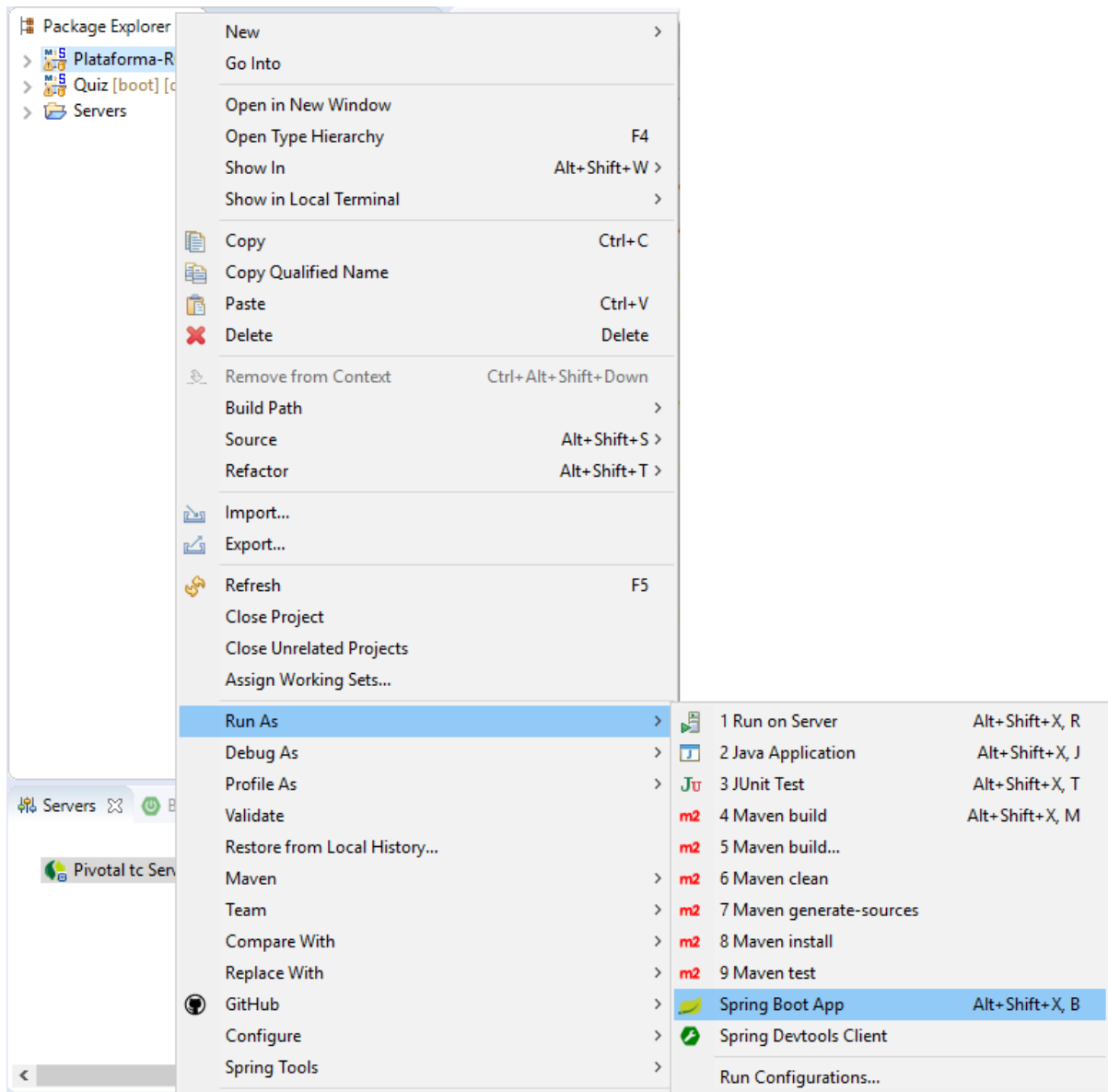


Figura 30. Ejecutar el proyecto.

Para ejecutar Sugar Engine:

Es necesario descargar el contenido del fichero *sugarengine* de la siguiente url <https://github.com/JorgeMerino/Plataforma-RGA/tree/Inicio/sugarengine>

Una vez descomprimido, ejecutar los siguientes comandos, desde un terminal de Windows:

- `vagrant up` (para levantar Vagrant)
- `vagrant ssh` (para acceder a la máquina virtual por ssh)
- `cp -r /vagrant/sugarengine .` (copiamos el directorio indicado al actual)
- `cd sugarengine`
- `./start.sh` (script que descarga y compila las imágenes de docker, la primera vez que se ejecuta tarda bastante tiempo)

Para detener la ejecución:

- `./stop.sh` (detener Sugar Engine)
- `exit` (para salir del entorno virtual)
- `vagrant halt` (para apagar la máquina virtual)

Anexo II: Manual de usuario

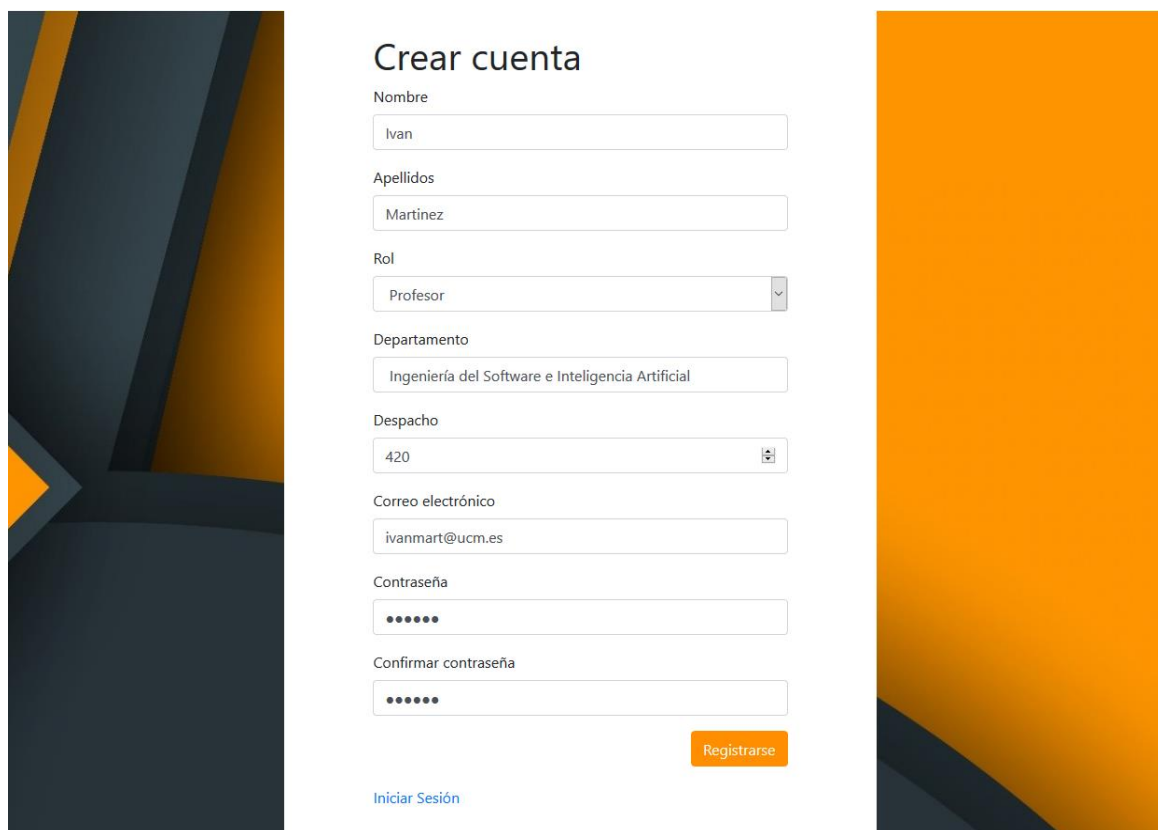
Después de haber seguido el manual del [Anexo I: Manual de instalación](#), se procederá a detallar el manual de usuario. Para ello, en primer lugar será necesario ejecutar la aplicación principal, la cual se mapea desde la URL <http://localhost:9000>. A continuación, se mostrarán las diferentes vistas de alumno y profesor de la aplicación.

Manual del profesor

La vista del profesor es importante, dado que ningún alumno puede tener acceso a sus asignaturas sin que el profesor le añada previamente.

Crear cuenta

Paso 1: Para registrarse debe introducir sus datos personales, su departamento y su despacho. También debe introducir un correo y contraseña, con los que se autenticará.



Crear cuenta

Nombre
Ivan

Apellidos
Martínez

Rol
Profesor

Departamento
Ingeniería del Software e Inteligencia Artificial

Despacho
420

Correo electrónico
ivanmart@ucm.es

Contraseña
•••••

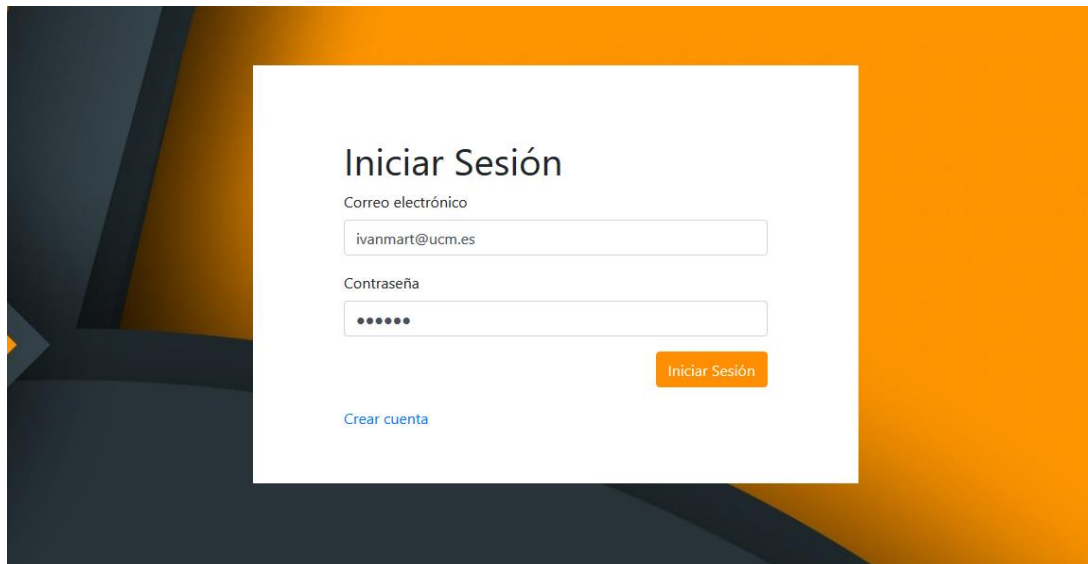
Confirmar contraseña
•••••

[Iniciar Sesión](#) [Registrarse](#)

Figura 31. Crear cuenta del profesor.

Iniciar sesión

Paso 2: Introduzca su correo electrónico y su contraseña para iniciar sesión, una vez ya se haya registrado.



The image shows a login form titled "Iniciar Sesión" (Login) on a white background with an orange border. It contains two input fields: "Correo electrónico" (Email) with the value "ivanmart@ucm.es" and "Contraseña" (Password) with masked characters. Below the password field is an orange button labeled "Iniciar Sesión" (Login). At the bottom left, there is a blue link labeled "Crear cuenta" (Create account).

Figura 32. Iniciar sesión.

Inicio

Paso 3: Haga clic en el botón "Nueva Asignatura" para añadir la asignatura que se impartirá durante el curso, rellenando los espacios con los datos que se le solicitan. Una vez creada aparecerá en la pantalla de inicio.



The image shows a dashboard titled "Mis Asignaturas" (My Courses) with a dark header and orange background. It features a table with columns: Id, Nombre, Grupo, Curso, and Acciones. The table is currently empty, displaying the message "No hay asignaturas que mostrar" (No courses to show). An orange button labeled "Nueva Asignatura" (New Course) is located at the bottom right of the table area.

Id	Nombre	Grupo	Curso	Acciones
No hay asignaturas que mostrar				

Figura 33. Inicio.

Nueva Asignatura [X]

Nombre
Bases de datos

Grupo
3ºA

Curso
2017/2018

Cerrar Añadir Asignatura

Figura 34. Añadir asignatura.

Mis Asignaturas				
Id	Nombre	Grupo	Curso	Acciones
3	Bases de datos	3ºA	2017/2018	[Icono]

Nueva Asignatura

Figura 35. Inicio.

Gamificación

Paso 4: Añada los alumnos que cursen su asignatura. Para ello, pulse en “Añadir Alumno” y vaya añadiendo los alumnos de uno en uno. Tenga en cuenta que ningún alumno podrá acceder a la asignatura si usted no lo ha agregado a esta lista. En la vista podrá observar los alumnos que ha introducido.



Figura 36. Añadir alumno.

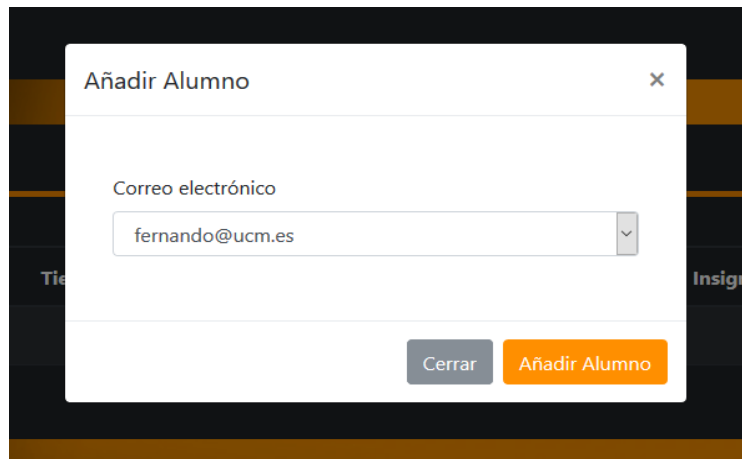


Figura 37. Añadir alumno.

Bases de datos						
Retos	Gamificación	Analíticas	Insignias			
#	Nombre	Puntuación	Tiempo Medio	Porcentaje	Insignias	Acciones
	Fernando Rívilla	0	0 s	0		
	Jorge Merino	0	0 s	0		
	Carlos Marron	0	0 s	0		
						Añadir Alumno

Figura 38. Vista de gamificación.

Retos

Paso 5: Una vez cumplimentada la pantalla gamificación proceda a crear el reto.

A través de esta pantalla usted será redirigido a Quiz donde podrá ir reformulando las preguntas y respuestas del reto. Continúa en el apartado [Manual del profesor, Quiz](#).

En la vista aparecerán todos los retos creados.

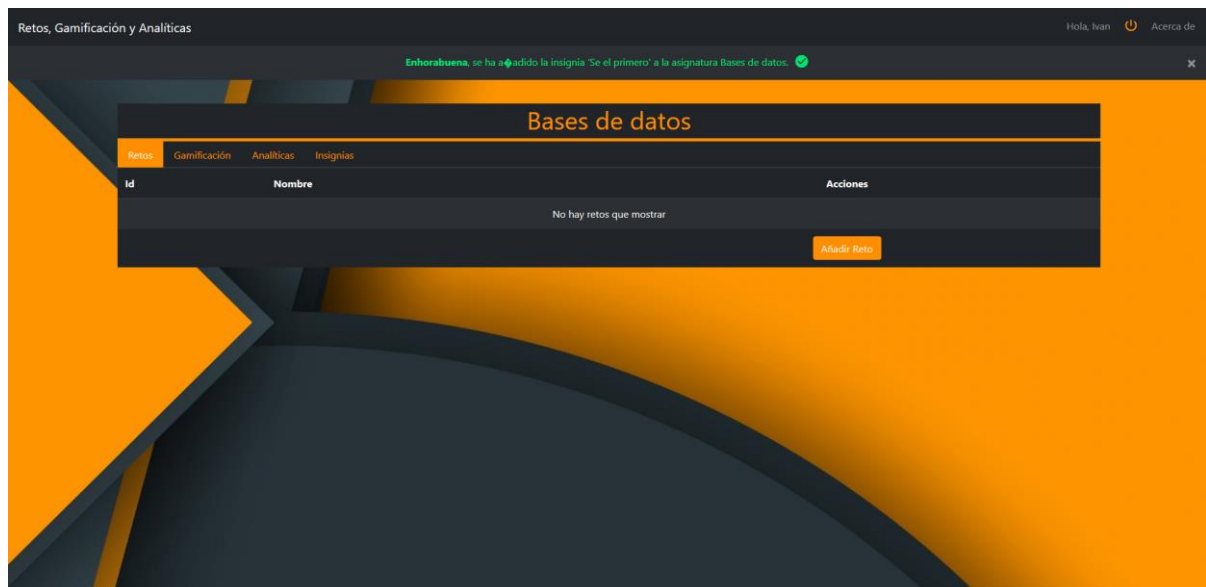


Figura 39. Insertar reto.

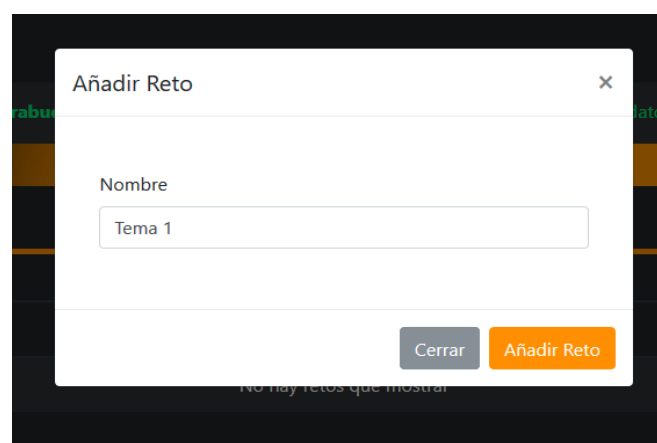


Figura 40. Insertar reto.

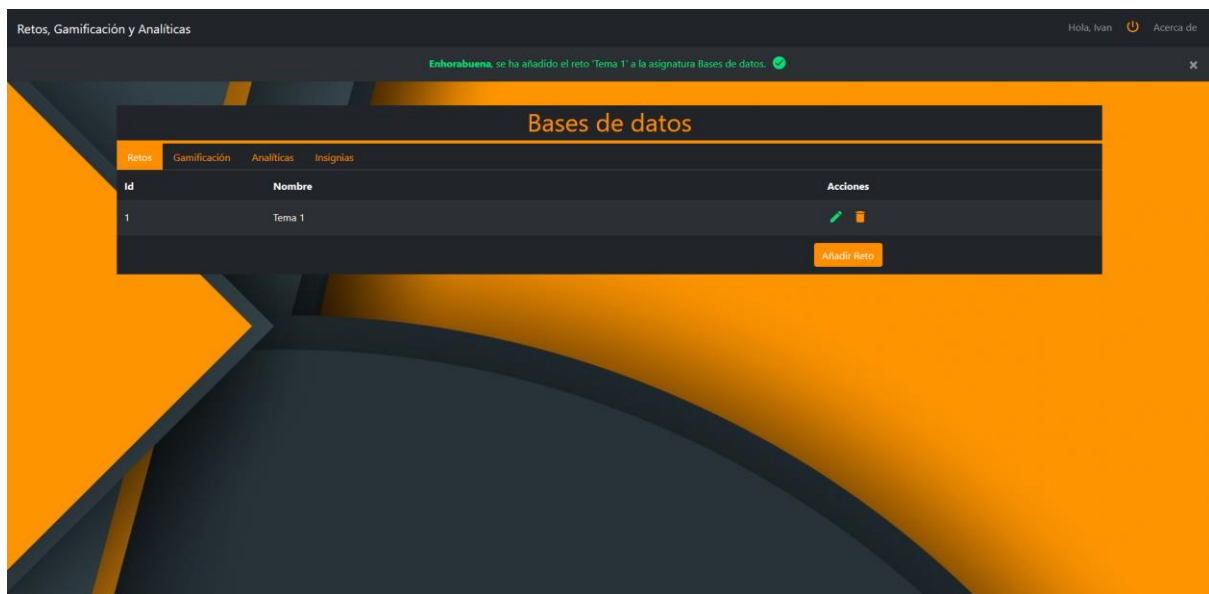


Figura 41. Insertar reto.

Insignias

Paso 6: Añadir las insignias que se entregarán al alumno. Estas pueden ser de diferentes categorías como oro, plata o bronce, dependiendo de la dificultad para conseguirla. Además, en este punto puede premiar al alumno que consiga alcanzar una meta. En la vista tendrá una lista con todas las insignias creadas.

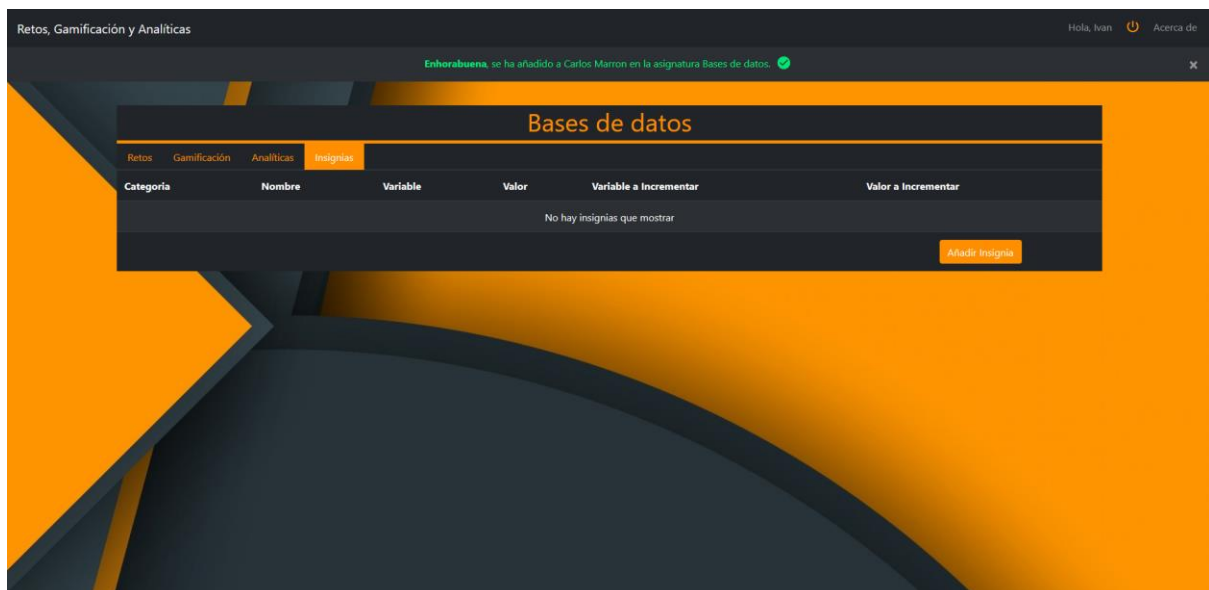


Figura 42. Añadir insignia.

Añadir Insignia

Nombre

Categoría

Variable

Valor

Comparacion

Variable a incrementar

Valor a incrementar

Cerrar
Añadir Insignia

Figura 43. Añadir insignia.

Retos, Gamificación y Analíticas

Hola, Ivan Acerca de

Enhorabuena, se ha añadido la insignia 'Se el primero' a la asignatura Bases de datos.

Bases de datos

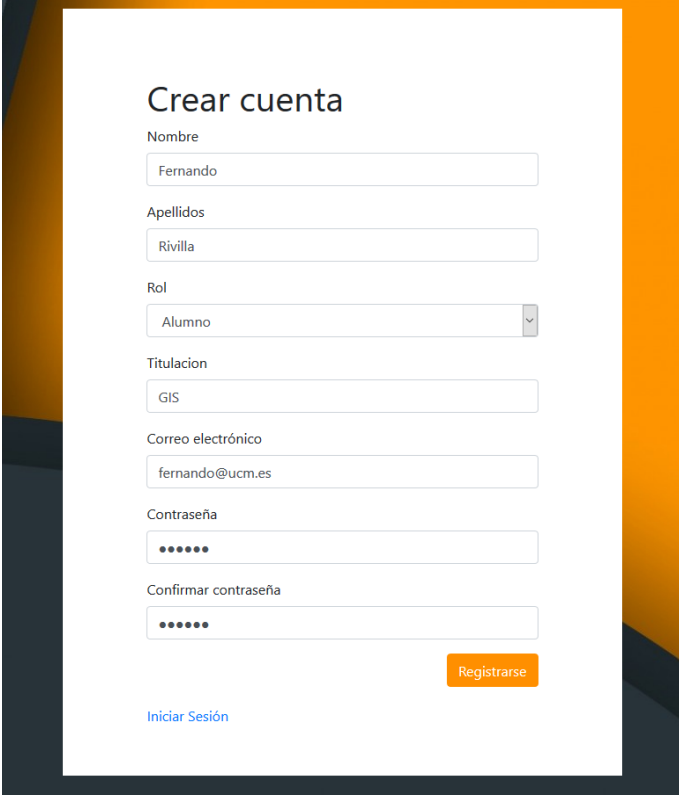
Retos	Gamificación	Analíticas	Insignias			
Categoría	Nombre	Variable	Valor	Variable a Incrementar	Valor a Incrementar	
	Se el primero	Puntuacion	1500	Puntuacion	100	Añadir Insignia

Figura 44. Añadir insignia.

Manual del alumno

Crear cuenta

Paso 1: Para registrarse debe introducir sus datos personales y su titulación. También debe introducir un email y contraseña, con los que posteriormente se autenticará.

El formulario, titulado "Crear cuenta", está diseñado para la creación de una cuenta de usuario. Incluye campos de entrada para "Nombre" (con el valor "Fernando"), "Apellidos" (con el valor "Rivilla"), "Rol" (un menú desplegable con "Alumno" seleccionado), "Titulación" (con el valor "GIS"), "Correo electrónico" (con el valor "fernando@ucm.es"), "Contraseña" y "Confirmar contraseña", ambas representadas por puntos. Un botón naranja "Registrarse" está situado a la derecha de los campos de contraseña. En la parte inferior izquierda, hay un enlace azul "Iniciar Sesión". El formulario está superpuesto sobre un fondo con franjas diagonales de color naranja y gris oscuro.

Crear cuenta

Nombre
Fernando

Apellidos
Rivilla

Rol
Alumno

Titulación
GIS

Correo electrónico
fernando@ucm.es

Contraseña
•••••

Confirmar contraseña
•••••

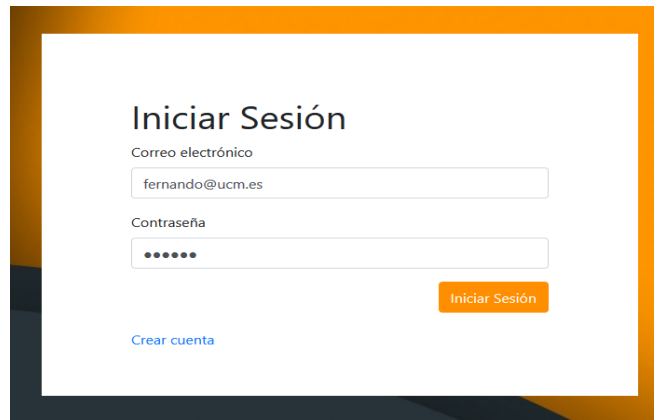
Registrarse

[Iniciar Sesión](#)

Figura 45. Crear cuenta del alumno.

Iniciar sesión

Paso 2: Introduzca su correo electrónico y su contraseña para iniciar sesión, una vez ya se haya registrado.



The image shows a login form titled "Iniciar Sesión". It has two input fields: "Correo electrónico" with the value "fernando@ucm.es" and "Contraseña" with masked characters. Below the password field is an orange button labeled "Iniciar Sesión". At the bottom left, there is a blue link labeled "Crear cuenta". The form is set against a white background with an orange border.

Figura 46. Iniciar sesión.

Inicio

En esta pantalla se le mostrarán las asignaturas en las que esté matriculado.



Figura 47. Inicio.

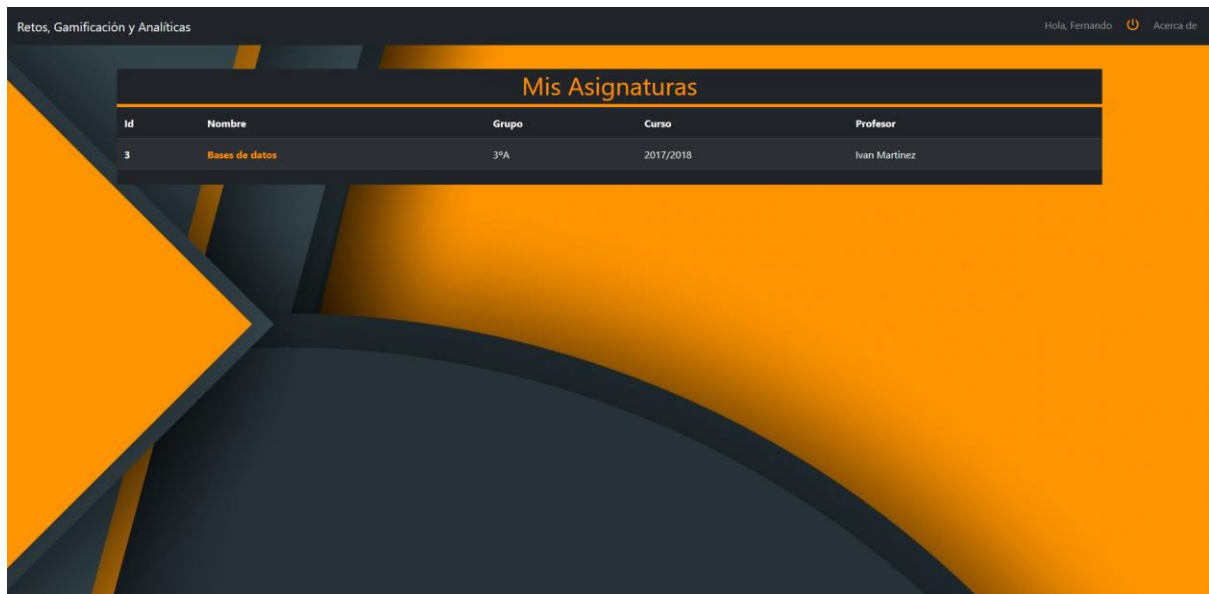


Figura 48. Inicio.

Gamificación


Aquí se podrán visualizar los datos relativos a la gamificación, de los alumnos matriculados. En esta pantalla se detalla la puntuación de cada participante, así como el tiempo medio de cada respuesta, el porcentaje y las insignias de cada uno.

Bases de datos					
Retos	Gamificación	Analíticas	Insignias		
#	Nombre	Puntuación	Tiempo Medio	Porcentaje	Insignias
	Fernando Rivilla	0	0 s	0	
	Jorge Merino	0	0 s	0	
	Carlos Marron	0	0 s	0	
Profesor de la asignatura: Ivan Martinez.			Departamento: Ingeniería del Software e Inteligencia Artificial. Despacho: 420		

Figura 49. Gamificación.

Retos

En caso de que el profesor haya lanzado algún reto, aparecerá la opción de “iniciar reto”, haciendo clic en ella se podrá comenzar a resolver. Continúa en [Manual del alumno, Quiz](#).



Bases de datos		
Retos	Gamificación	Insignias
Id	Nombre	Enlace
1	Tema 1	Iniciar reto

Profesor de la asignatura: Ivan Martinez. Departamento: Ingeniería del Software e Inteligencia Artificial. Despacho: 420

Figura 50. Retos.

Insignias

Para que usted sepa la categoría del reto, los puntos necesarios para superarlo y el premio que obtendrá, podrá consultar las insignias creadas por el profesor.



Bases de datos					
Retos	Gamificación	Analíticas	Insignias		
Categoría	Nombre	Variable	Valor	Variable a Incrementar	Valor a Incrementar
👤	Se el primero	Puntuacion	1500	Puntuacion	100

Profesor de la asignatura: Ivan Martinez. Departamento: Ingeniería del Software e Inteligencia Artificial. Despacho: 420

Figura 51. Insignias.

Manual del profesor, Quiz

Crear reto

Cuando haga clic en Modificar Reto en la pantalla Reto de la Plataforma RGA, será redirigido a Quiz, donde podrá crear o modificar un reto. Para ello:

- Pulse “añadir pregunta” y redacte la pregunta que desee formular.
- Escriba las opciones, seleccione la opción correcta y seleccione el tiempo que el alumno tendrá para resolverla.
- Haga clic en “añadir respuesta”.

Una vez haya añadido todas las preguntas que desee, podrá:

- Dejarlo preparado para lanzarlo posteriormente.
- Lanzarlo para que sus alumnos lo inicien.

En todo momento podrá añadir, eliminar o modificar cualquier pregunta.



Figura 52. Añadir preguntas.

Reto - Tema 1

Pregunta

Con que sentencia se inserta en una base de datos

Opción	Correcta	Eliminar
<div>SELECT</div>	<input type="radio"/>	
<div>INSERT INTO</div>	<input checked="" type="radio"/>	

Tiempo de Respuesta: 20 segundos

Atrás

Añadir Opción

Guardar Pregunta

Figura 53. Añadir pregunta.



Figura 54. Añadir preguntas.

Inicio del reto

Una vez lanzado el reto pasará a una pantalla vacía, esta será la “Sala de Espera” donde irán apareciendo los alumnos según vayan entrando al reto. Cuando determine, podrá iniciar el reto pulsando “Comenzar Reto”.



Figura 55. Sala de espera.



Figura 56. Sala de espera.

Reto

Dado que se trata de un sistema de retos en vivo, la siguiente pantalla será la que tenga que mostrar a los alumnos, en ella aparece el nombre del reto, la pregunta, las diferentes opciones de respuesta y el tiempo restante.

Deberá pasar las preguntas una a una. Los alumnos podrán verlas mediante la pantalla a la cual esté conectado. Una vez finalice el reto, se mandarán todas las respuestas a la Plataforma RGA para que usted pueda conocer todos los resultados.

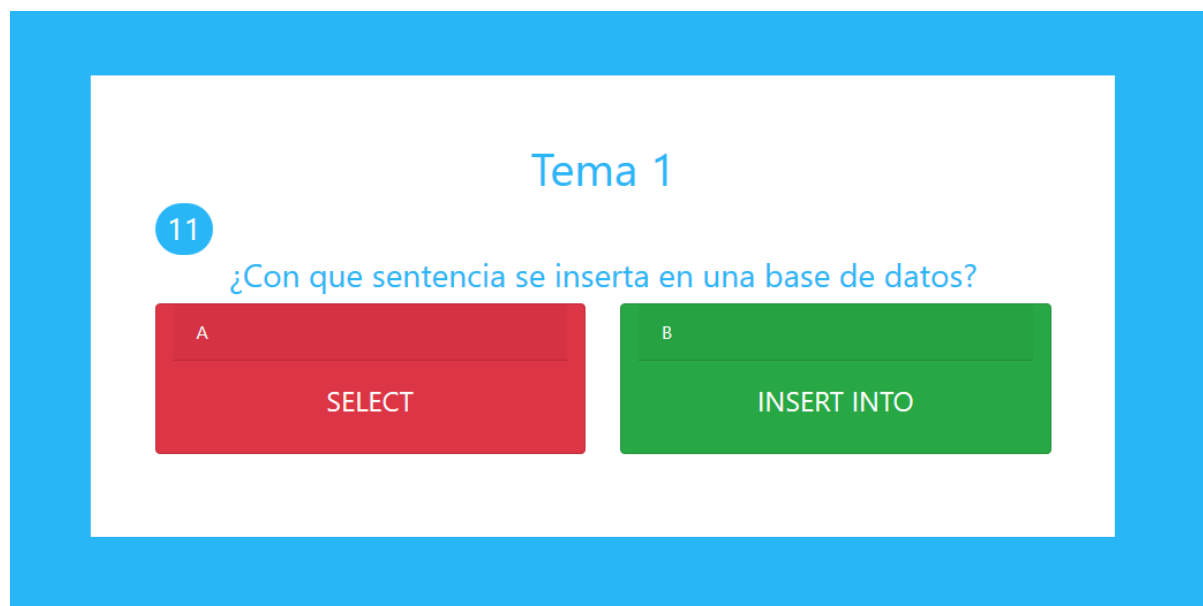


Figura 57. Reto.

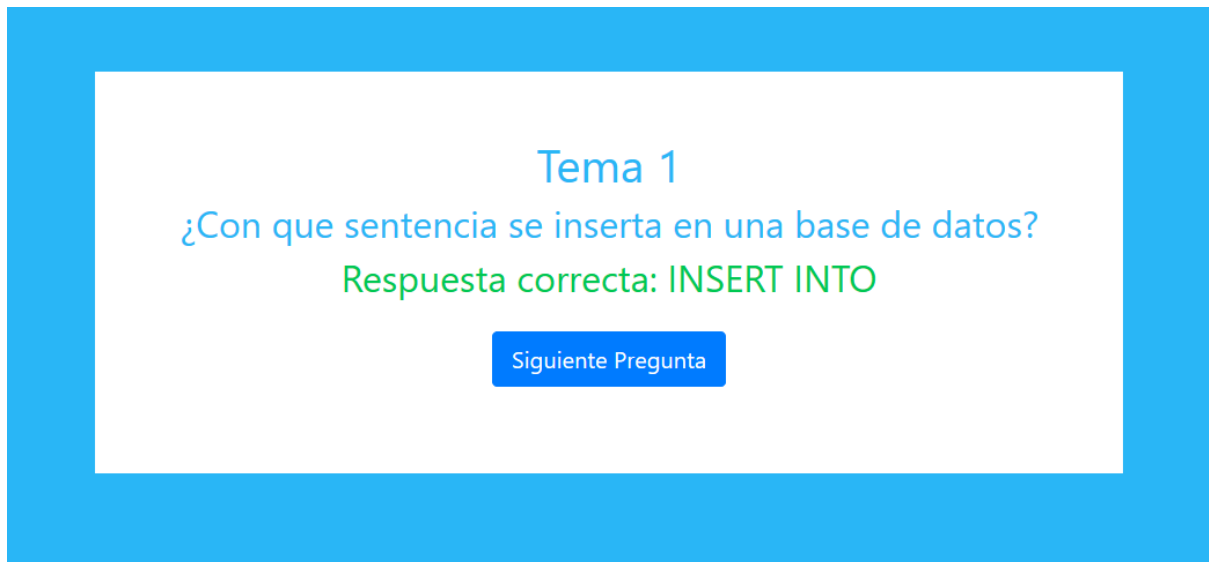


Figura 58. Fin de pregunta del reto.

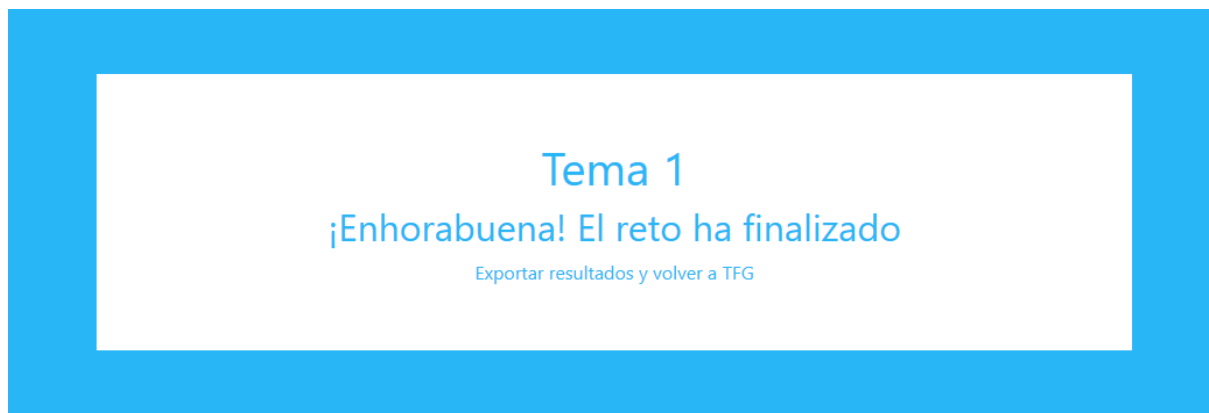


Figura 59. Fin del reto.

Manual del alumno, Quiz

Inicio de reto

Una vez pulsado “Iniciar Reto” en la Plataforma RGA, será redirigido a Quiz. Visualizará una pantalla en la cual deberá escribir su apodo y pasará a la sala de espera, donde se irán añadiendo los alumnos y esperará hasta que el profesor inicie el reto.



Figura 60. Introducir el apodo.



Figura 61. Sala de espera.



Figura 62. Sala de espera.

Resolver el reto

En su dispositivo aparecerán las opciones de respuesta que usted puede pulsar. Al mismo tiempo podrá ver las preguntas en la pantalla que el profesor muestre, en esta pantalla también podrá ver el nombre de cada opción y el tiempo que tiene para contestar. Una vez haya pulsado una opción verá si ha acertado o ha fallado.

Cuando el reto termine podrá volver a la Plataforma RGA.

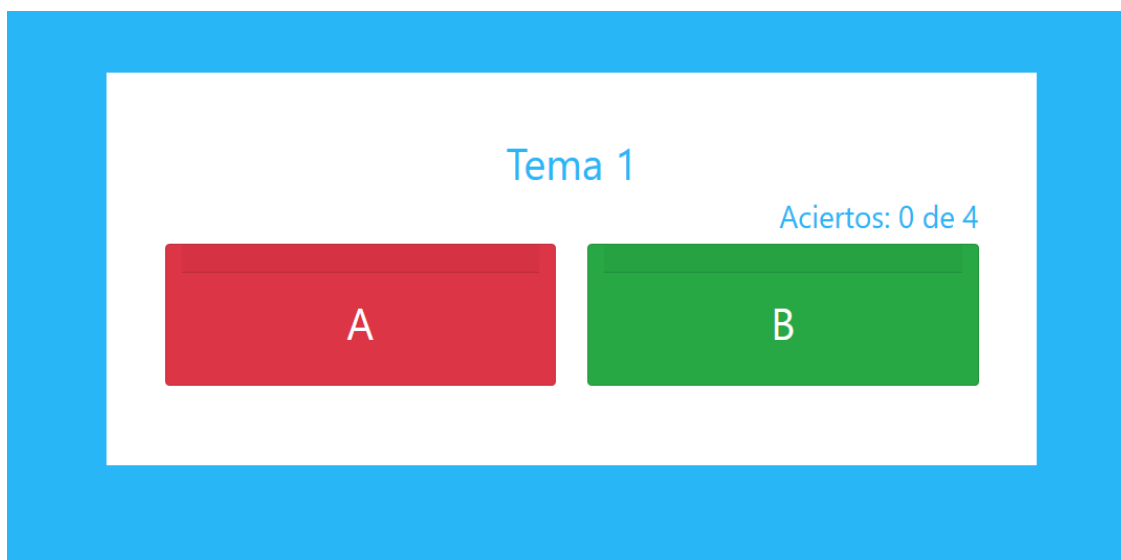


Figura 63. Resolución del reto.

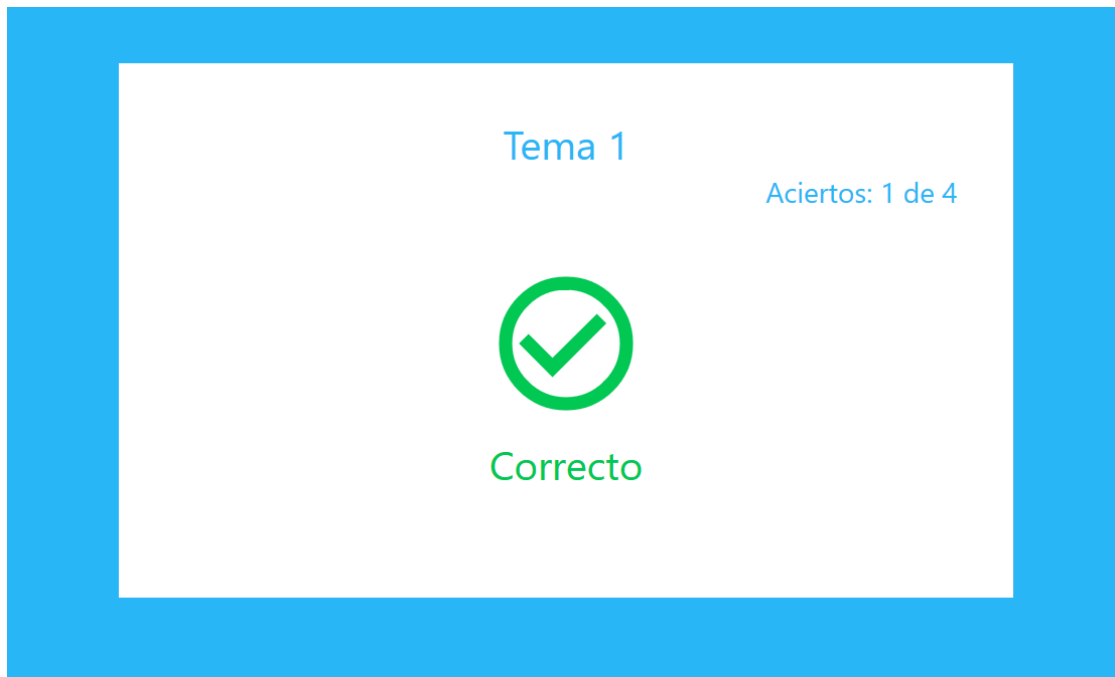


Figura 64. Respuesta correcta.

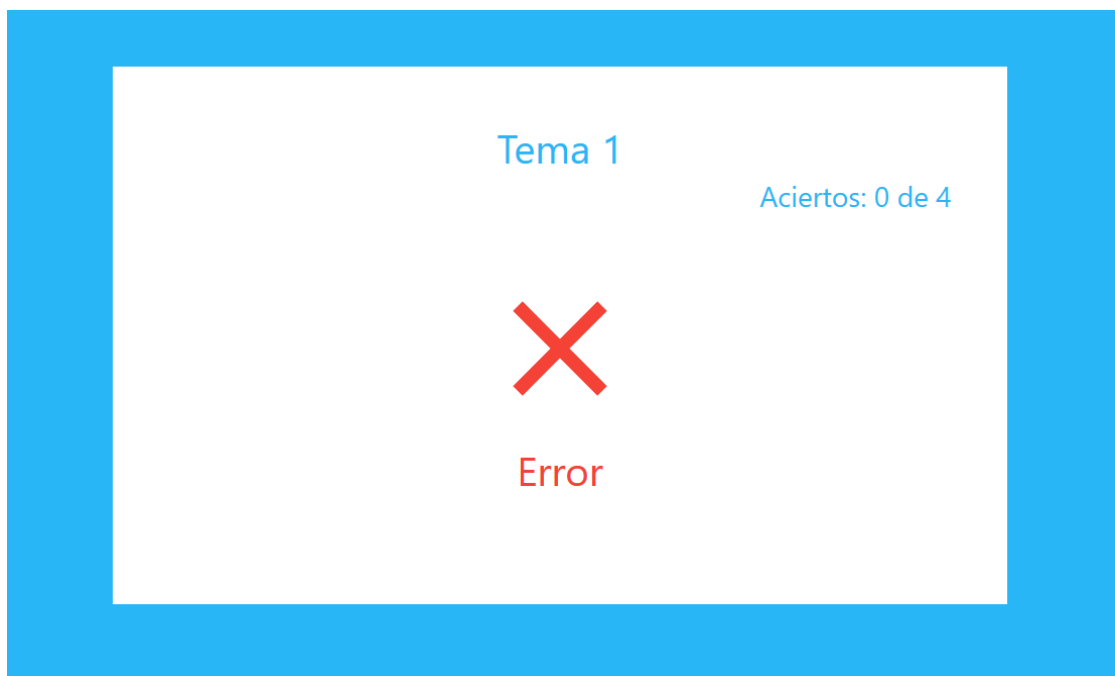


Figura 65. Respuesta incorrecta.

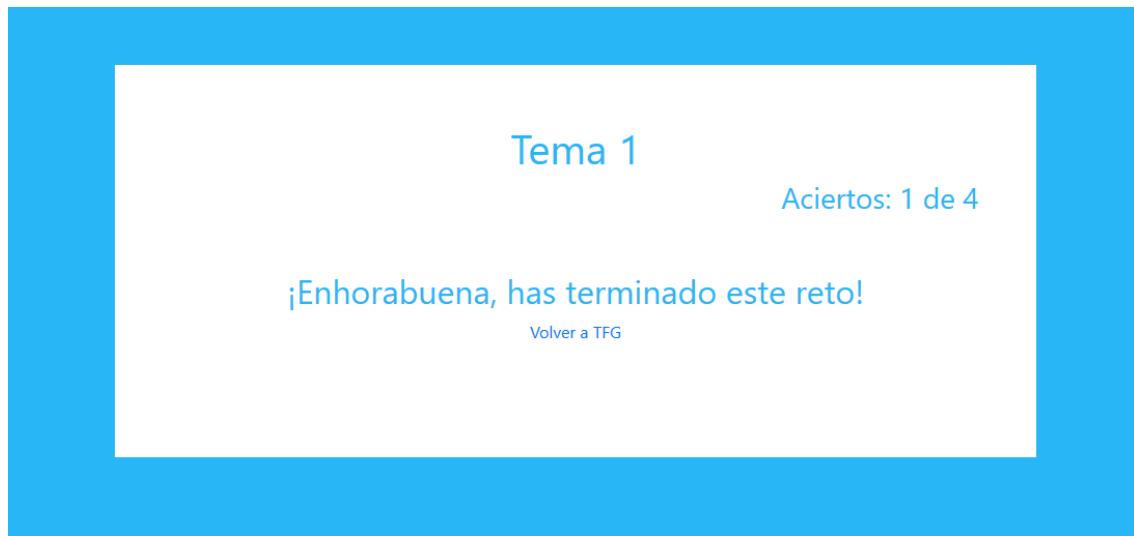


Figura 66. Fin del reto.

Ranking y analíticas

Para finalizar, tanto en la vista de profesor como en la del alumno, dentro de la Plataforma RGA, podrá ver el ranking de la clase una vez realizado el reto y la tabla de analíticas con las respuestas seleccionadas por cada alumno participante en el reto.

Bases de datos					
Retos	Gamificación	Analíticas	Insignias		
#	Nombre	Puntuación	Tiempo Medio	Porcentaje	Insignias
	Jorge Merino	60	14 s	150	
	Fernando Rivilla	30	13 s	150	
	Carlos Marron	30	4 s	75	
Profesor de la asignatura: Ivan Martinez.			Departamento: Ingeniería del Software e Inteligencia Artificial. Despacho: 420		

Figura 67. Ranking o Leaderboard.

Bases de Datos

Retos

Gamificación

Analíticas

Insignias

Reto:

Prueba Inicial

Mostrar:

Tabla de contenido de resultados

Tabla de contenido de resultados

Alumno	Reto	Pregunta	Respuesta	Correcta	Tiempo
Luis Rodríguez	Prueba Inicial	Indica cuál no es un lenguaje de BD Relacionales	MongoDB	✓	8.0 s
Luis Rodríguez	Prueba Inicial	Para seleccionar todas las filas devueltas de una consulta utilizaremos:	SELECT	✗	11.0 s
Luis Rodríguez	Prueba Inicial	¿Qué consulta usaremos para borrar la tabla Empleados?	DELETE TABLE Empleados	✗	6.0 s
Luis Rodríguez	Prueba Inicial	¿Cuál es el conector de BD de Java?	JDBC	✓	9.0 s
Luis Rodríguez	Prueba Inicial	Para actualizar utilizaremos:	UPDATE	✓	6.0 s
Carolina Sánchez	Prueba Inicial	Indica cuál no es un lenguaje de BD Relacionales	MongoDB	✓	9.0 s
Carolina Sánchez	Prueba Inicial	Para seleccionar todas las filas devueltas de una consulta utilizaremos:	SELECT *	✓	12.0 s
Carolina Sánchez	Prueba Inicial	¿Qué consulta usaremos para borrar la tabla Empleados?	DELETE TABLE Empleados	✗	7.0 s
Carolina Sánchez	Prueba Inicial	¿Cuál es el conector de BD de Java?	CBDJ	✗	8.0 s
Carolina Sánchez	Prueba Inicial	Para actualizar utilizaremos:	UPDATE	✓	14.0 s
Lorena Sanz	Prueba Inicial	Indica cuál no es un lenguaje de BD Relacionales	Oracle SQL	✗	18.0 s
Lorena Sanz	Prueba Inicial	Para seleccionar todas las filas devueltas de una consulta utilizaremos:	SELECT *	✓	4.0 s
Lorena Sanz	Prueba Inicial	¿Qué consulta usaremos para borrar la tabla Empleados?	DROP TABLE Empleados	✓	3.0 s
Lorena Sanz	Prueba Inicial	¿Cuál es el conector de BD de Java?	JDBC	✓	5.0 s
Lorena Sanz	Prueba Inicial	Para actualizar utilizaremos:	UPDATE	✓	15.0 s

Figura 68. Tabla analítica de resultados.

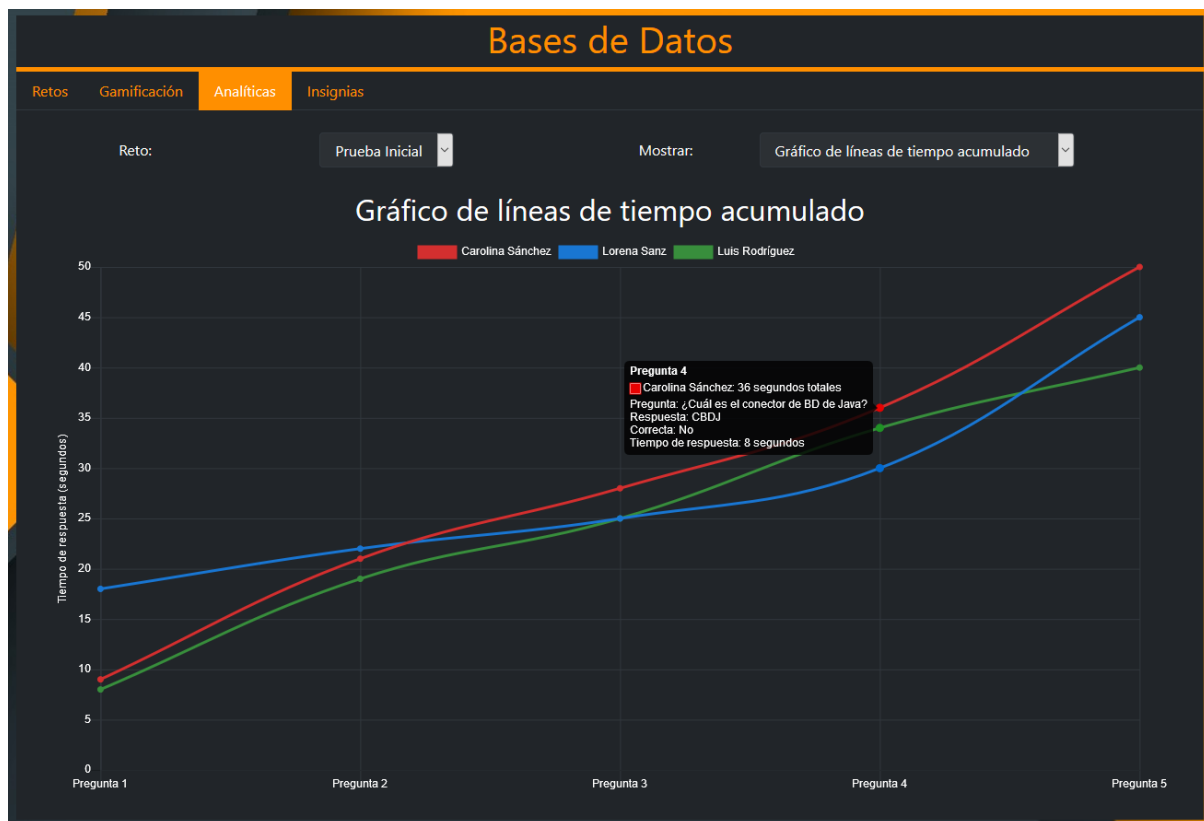


Figura 69. Gráfico analítico de tiempo.

Anexo III: Referencia de la API de Plataforma RGA

A continuación se mostrará el contenido del fichero .yaml que genera la aplicación de Swagger Editor. Donde quedan detalladas las características de las diferentes peticiones REST a la API de Plataforma RGA.

```
swagger: '2.0'
info:
  description: >-
    En este documento quedan recogidas las peticiones API REST de la
    Plataforma
    RGA, del Trabajo de Fin de Grado con nombre 'Plataforma de
    Retos,
    Gamificación y Analíticas'.
  version: 1.0.0
  title: API RGA
host: 'localhost:9000'
basePath: /api
tags:
  - name: reto
    description: Retos de Plataforma RGA
  - name: usuario
    description: Usuarios de Plataforma RGA
schemes:
  - http
paths:
  '/reto/{idReto}/activar-disponibilidad':
    post:
      tags:
        - reto
      summary: Activa la disponibilidad de un reto
      description: ''
      operationId: activarDisponibilidad
      consumes:
        - multipart/form-data
      produces:
        - application/json
      parameters:
        - name: idReto
          in: path
          description: ID del reto que se activará la disponibilidad
          required: true
          type: integer
          format: int64
      responses:
        '200':
          description: successful operation
          schema:
            $ref: '#/definitions/Reto'
```

```

'/reto/{idReto}/cancelar-disponibilidad':
  post:
    tags:
      - reto
    summary: Cancela la disponibilidad de un reto
    description: ''
    operationId: cancelarDisponibilidad
    consumes:
      - multipart/form-data
    produces:
      - application/json
    parameters:
      - name: idReto
        in: path
        description: ID del reto que se cancelará la
disponibilidad
        required: true
        type: integer
        format: int64
    responses:
      '200':
        description: successful operation
        schema:
          $ref: '#/definitions/Reto'
'/reto/{idReto}/importar':
  post:
    tags:
      - reto
    summary: Importa los resultados de un reto
    description: ''
    operationId: importar
    consumes:
      - multipart/form-data
    produces:
      - application/json
    parameters:
      - name: idReto
        in: path
        description: ID del reto del que se importarán los
resultados
        required: true
        type: integer
        format: int64
      - in: body
        name: petition
        required: true
        schema:
          $ref: '#/definitions/Reto'
    responses:
      '200':
        description: successful operation
        schema:
          $ref: '#/definitions/Reto'
'/reto/{idReto}/ir-a-asignatura':
  get:

```

```

tags:
  - reto
summary: Redirige a la asignatura de un reto
description: ''
operationId: irAAsignatura
consumes:
  - multipart/form-data
produces:
  - application/json
parameters:
  - name: idReto
    in: path
    description: ID del reto al que se redirigirá a su
asignatura
    required: true
    type: integer
    format: int64
responses:
  '200':
    description: successful operation
    schema:
      $ref: '#/definitions/Reto'
/comprobar-usuario:
get:
  tags:
    - usuario
summary: Comprueba un usuario
description: ''
operationId: comprobarUsuario
produces:
  - application/json
consumes:
  - multipart/form-data
parameters:
  - name: idUsuario
    in: formData
    description: ID del usuario que se comprobará
    required: true
    type: string
  - name: token
    in: formData
    description: Token del usuario que se comprobará
    required: true
    type: string
responses:
  default:
    description: successful operation
    schema:
      $ref: '#/definitions/Usuario'
definitions:
  Reto:
    type: object
    properties:
      id:
        type: integer

```

```

    format: int64
  nombre:
    type: string
xml:
  name: Reto
Usuario:
  type: object
  properties:
    id:
      type: integer
      format: int64
    nombre:
      type: string
    token:
      type: string
xml:
  name: Usuario

```

Como ejemplo, se mostrará la petición REST de activar disponibilidad de un reto, en la siguiente figura.

POST

/reto/{idReto}/activar-disponibilidad

Activa la disponibilidad de un reto

Parameters

Try it out

Name	Description
idReto * required integer(\$int64) (path)	ID del reto que se activará la disponibilidad

Responses

Response content type

application/json

Code	Description
200	<div>successful operation</div> <div> Example Value Model <pre>{ "id": 0, "nombre": "string" }</pre> </div>

Figura 70. Activar disponibilidad.

Anexo IV: Referencia de la API de Quiz

A continuación se mostrará el contenido del fichero .yaml que genera la aplicación de Swagger Editor. Donde quedan detalladas las características de las diferentes peticiones REST a la API de Quiz.

```
swagger: '2.0'
info:
  description: >-
    En este documento quedan recogidas las peticiones API REST de la
    aplicación
    Quiz, del Trabajo de Fin de Grado con nombre 'Plataforma de
    Retos,
    Gamificación y Analíticas'.
  version: 1.0.0
  title: API Quiz
host: 'localhost:8000'
basePath: /api
tags:
  - name: reto
    description: Retos de la aplicación Quiz
schemes:
  - http
paths:
  '/reto/{idReto}/crear':
    post:
      tags:
        - reto
      summary: Crea un reto
      description: ''
      operationId: crearReto
      consumes:
        - multipart/form-data
      produces:
        - application/json
      parameters:
        - name: idReto
          in: path
          description: ID del reto que se creará
          required: true
          type: integer
          format: int64
      responses:
        '200':
          description: successful operation
          schema:
            $ref: '#/definitions/Reto'
  '/reto/{idReto}/resultados':
    get:
      tags:
        - reto
      summary: Obtiene los resultados de un reto
```

```

    description: ''
    operationId: obtenerResultados
    consumes:
      - multipart/form-data
    produces:
      - application/json
    parameters:
      - name: idReto
        in: path
        description: ID del reto del que se obtendrán los
resultados
        required: true
        type: integer
        format: int64
    responses:
      '200':
        description: successful operation
        schema:
          $ref: '#/definitions/Reto'
  /reto/eliminar:
    post:
      tags:
        - reto
      summary: Elimina un reto
      description: ''
      operationId: eliminarReto
      produces:
        - application/json
      parameters:
        - in: body
          name: id
          description: ID del reto que se eliminará
          required: true
          schema:
            $ref: '#/definitions/Reto'
      responses:
        default:
          description: successful operation
definitions:
  Reto:
    type: object
    properties:
      id:
        type: integer
        format: int64
      nombre:
        type: string
    xml:
      name: Reto

```

Como ejemplo, se mostrará la petición REST de crear un reto, en la siguiente figura.

POST

/reto/{idReto}/crear

Crea un reto

Parameters

Try it out

Name	Description
idReto * required integer(\$int64) (path)	ID del reto que se creará

Responses

Response content type

application/json

Code	Description
200	<div>successful operation</div> <div>Example Value Model</div> <div><pre>{ "id": 0, "nombre": "string" }</pre></div>

Figura 71. Crear reto.

